

UNIVERSIDAD NACIONAL DEL SUR

TESIS DE DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

**Aplicación de mecanismos reactivos y argumentativos
para la búsqueda temática en redes P2P**

Ana Lucía Nicolini

BAHÍA BLANCA

ARGENTINA

2017

Prefacio

Esta Tesis es presentada como parte de los requisitos para optar al grado académico de Doctor en Ciencias de la Computación, de la Universidad Nacional del Sur, y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otras. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el Departamento de Ciencias de la Computación, durante el período comprendido entre el 1 de abril de 2013 y el 30 de septiembre de 2017, bajo la dirección de la Dra. Ana G. Maguitman y la co Dirección del Dr. Carlos I. Chesñevar, Profesores Asociados del Departamento de Ciencias de la Computación.

Lic. Ana Lucía Nicolini

`aln@cs.uns.edu.ar`

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

UNIVERSIDAD NACIONAL DEL SUR

Bahía Blanca, 7 de Octubre de 2017.



UNIVERSIDAD NACIONAL DEL SUR
Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el .../.../..., mereciendo
la calificación de(.....)

Agradecimientos

Durante el último año de mi carrera de grado se me hacía difícil divisar un futuro no tan lejano. La investigación era un área de la cual había escuchado hablar pero nunca me imaginé llegar a involucrarme. Gracias a mis directores, la Dra. Ana Maguitman y el Dr. Carlos Chesñear es que pude comprender qué era realmente investigar y me demostraron que yo era capaz de hacerlo. Fueron mis guías en los momentos de incertidumbre y me enseñaron a nunca bajar los brazos y a valorar el trabajo realizado. En el transcurso de mis años como becaria doctoral he tropezado con distintos contratiempos pero ellos siempre me mostraron la forma de seguir adelante. Es por todo esto que deseo expresar en primera instancia mi agradecimiento sincero hacia ellos ya que reconozco su esfuerzo para formarme como científica brindándome su tiempo, apoyo, conocimiento y sobretodo paciencia. El mundo de la investigación se encuentra lleno de pequeños problemas que hay que resolver día a día y para ello siempre conté con el apoyo incondicional de Dr. Carlos Lorenzetti, quien dedicó horas y horas de su tiempo exclusivamente para ayudarme cada vez que algo no funcionaba de la manera esperada. Sin ellos, llegar hasta esta instancia tan importante en mi vida no hubiese sido posible.

Los momentos de incertidumbre fueron varios pero siempre estuvieron a mi lado mis compañeros de la sala de becarios con quienes compartimos muchas experiencias, algunas felices y otras no tanto. De estas últimas aprendí que siempre es mejor cuando se puede compartir el momento poco feliz que se está atravesando con alguien que quizás ya lo vivió. Ellos siempre generaron el clima de trabajo que uno realmente necesita para trabajar en forma activa, un clima ameno y muy placentero. En especial quiero agradecerles a Julieta Dussaut, Cecilia Baggio y Carlos Teze por ser tan buenos amigos y compañeros.

Además agradezco a todos los integrantes del Departamento de Ciencias e Ingeniería de la Computación (DCIC) y a los miembros del Instituto de Ciencias e Ingeniería de la Computación (ICIC) por darme la posibilidad de trabajar junto a ellos todos estos años.

Agradezco especialmente a los docentes que fueron profesores míos durante mi carrera de grado y a todos aquellos con los que tuve la oportunidad de trabajar como ayudante de cátedra, en especial al Dr. Luciano Tamargo por brindarme plenamente su apoyo y confianza. También quiero agradecer al personal directivo, administrativo y técnico de ambas entidades, quienes desarrollan una silenciosa pero excelente y eficaz tarea en lo que respecta a trámites administrativos y servicios de mantenimiento.

Agradezco a toda mi familia, quienes siempre me guiaron por el camino del estudio, la responsabilidad, los valores y la honestidad. En especial dedico este trabajo a mis padres Lucrecia y Julio quienes me educaron con mucho amor y me enseñaron la importancia de trabajar esmeradamente. También agradezco profundamente a mi gran amor, Rodrigo, por darme la posibilidad de compartir la vida con él brindándome su amor incondicional que siempre fue un pilar fundamental para superar aquellos momentos duros que me han tocado transitar.

Si bien la carrera científica está encuadrada en un ámbito académico, las relaciones sociales activas me han ayudado a desarrollarme plenamente como profesional. Por este motivo no quiero dejar de agradecer a todo el equipo del Establecimiento Hípico LCC, quienes me brindaron su apoyo tanto desde lo deportivo como desde lo profesional. De ellos aprendí que siempre se puede mejorar mirando hacia adelante, sin bajar los brazos y sobre todo con mucho trabajo duro. En especial quiero agradecer por su apoyo incondicional a mis instructores y amigos Yanina y Juan Pablo y a mi compañera del día a día a quién aprecio mucho, Valentina.

Por último, agradezco al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) por haber confiado en mí financiando mediante becas mi doctorado en forma completa. Esta tesis no hubiera sido posible sin el apoyo económico de esta institución de la que me enorgullece formar parte.

¡Gracias a todos!

Anita

Resumen

Las redes peer-to-peer descentralizadas son sistemas distribuidos compuestos de nodos en los que cada uno de estos participantes puede comunicarse del mismo modo con cualquier otro de los participantes tanto bajo el rol de servidor de contenido como de demandante del mismo. En este escenario, los algoritmos de búsqueda temática deben conducir a y beneficiarse de la aparición de comunidades semánticas que son el resultado de la interacción entre los participantes. Varios estudios han demostrado la importancia de la selección de vecinos para el ruteo de consultas en redes peer-to-peer. La mayoría de los algoritmos existentes de ruteo de consultas aplican un enfoque reactivo, lo que conduce a una serie de limitaciones. Esta investigación propone aplicar un enfoque razonado basado en la argumentación que otorga a cada nodo capacidades de razonamiento. El framework resultante, llamado *ArgP2P*, es descrito formalmente y evaluado empíricamente en el desarrollo de esta Tesis. Un cuantioso número de simulaciones basadas en datos realistas revelan el potencial de *ArgP2P* para hacer frente a diferentes problemas comúnmente observados en sistemas peer-to-peer reactivos, como el problema de las comunidades cerradas y el manejo de la congestión.

Abstract

Decentralized peer-to-peer networks are distributed systems of peers where each peer can communicate as an equal to any other peer, serving content as well as requesting it. In this scenario, thematic search algorithms should lead to and benefit from the emergence of semantic communities that are the result of the interaction among participants. A number of studies have demonstrated the importance of neighbor selection for query routing in peer-to-peer networks. Most existing query-routing algorithms apply a reactive approach, which leads to a number of limitations. This research proposes to apply a reasoned approach based on argumentation that endows each node with reasoning capabilities. The resulting framework, called *ArgP2P*, is formally described and empirically evaluated. Simulations based on realistic data demonstrate the potential of *ArgP2P* to deal with different problems commonly observed in reactive peer-to-peer systems, such as the closed communities problem and management of congestion.

Publicaciones surgidas de esta Tesis

En revistas internacionales:

- Nicolini, Ana L., Lorenzetti, Carlos M., Maguitman, Ana G., Chesñevar, Carlos I. (2017). Intelligent Algorithms for Improving Communication Patterns in Thematic P2P Search, *Information Processing and Management*, Vol. 53 pp. 388–404.

En workshops internacionales:

- Nicolini A. L, Maguitman A.G., Chesñevar C. I. (2015), ArgP2P: An argumentative approach for intelligent query routing in P2P networks. The 2015 International Workshop on Theory and Applications of Formal Argument (TAFA). International Joint Conference on Artificial Intelligence (IJCAI). Buenos Aires, Argentina. *Lecture Notes in Computer Science*, Vol. 9524. Springer.

En congresos nacionales:

- Nicolini A. L, Lorenzetti C., Maguitman A.G., Chesñevar C. I. (2013), Intelligent Algorithms for Reducing Query Propagation in Thematic P2P Search. XIX Congreso Argentino de Ciencias de la Computación (CACIC), pp. 71–79. 2013. Mar del Plata, Argentina.

Tesinas

- Nicolini A. L, Luna M. N, Ciarrochi M, 2017. Algoritmos para el Acceso a Información Temática en Entornos Distribuidos. Tesis de licenciatura en ciencias de la computación. Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur. Bahía Blanca, Argentina.

Primer Ronda de Revisión

- Nicolini, Ana L., Lorenzetti, Carlos M., Maguitman, Ana G., Chesñevar, Carlos I. (2017). Semantic issues for intelligent query routing in P2P networks: a survey. Enviado a Knowledge and Information Systems (KAIS). Springer.
- Nicolini, A.L., Lorenzetti, C.M., Maguitman, A.G. and Chesñevar, C.I (2017). Boosting Performance in Thematic Search Query Routing in P2P Networks through Argumentation. Enviado a Information Processing Letters (IPL). Elsevier.

Índice general

1. Introducción	1
1.1. Motivación	2
1.2. Objetivos	4
1.3. Resumen de las técnicas propuestas y contribuciones	5
1.4. Organización de la Tesis	7
2. Fundamentos	9
2.1. Características y clasificación de las redes P2P	10
2.1.1. Clasificación	11
2.1.2. Aspectos semánticos	13
2.1.3. Clasificación de los algoritmos	14
2.1.4. Características topológicas de una red lógica	15
2.2. Razonamiento no-monótono y argumentación rebatible	19
2.2.1. Argumentación basada en suposiciones	23
2.3. Resumen del capítulo	28
3. Algoritmos de Ruteo: una visión en perspectiva	29
3.1. Algoritmos de ruteo	30
3.1.1. Algoritmos de ruteo en redes P2P estructuradas	30
3.1.2. Algoritmos de ruteo en redes P2P semi-estructuradas o débilmente estructuradas	33

3.1.3. Algoritmos de ruteo en redes P2P no estructuradas	35
3.2. Análisis comparativo	43
3.3. Resumen del capítulo	45
4. Algoritmos de ruteo en redes P2P	47
4.1. Algoritmos inteligentes para mejorar patrones de comunicación en redes P2P	48
4.1.1. Principales características de los algoritmos	55
4.2. ArgP2P: combinando comportamientos reactivos y argumentativos	56
4.2.1. Formalizando la noción de red P2P	59
4.2.2. El framework ArgP2P	60
4.2.3. Caso de estudio	66
4.3. Resumen del capítulo	72
5. Evaluación	73
5.1. Evaluación de los algoritmos inteligentes	73
5.1.1. Configuraciones de la simulación	74
5.1.2. Surgimiento de patrones de comunicación y estructuras con topología de mundo pequeño	76
5.1.3. Evaluación de la escalabilidad	81
5.1.4. Distribución de grados y descomposición en k-núcleos	82
5.1.5. Topología de las comunidades semánticas	85
5.2. Evaluación del sistema Argumentativo	87
5.3. Evaluación integral	90
5.4. Resumen del capítulo	98
6. Conclusiones y trabajo a futuro	101
6.1. Conclusiones	101
6.2. Trabajo a futuro	105
A. Glosario	109

Índice de figuras

1.1. Elementos conceptuales que caracterizan una red P2P	3
2.1. Ejemplo de red P2P no estructurada.	12
2.2. Ejemplo de una red con topología de mundo pequeño.	17
2.3. Ejemplos del grado de un nodo. Grado 1: 5, 17, 18 y 19. Grado 2: 2, 4, 6, 7, 11, 13, 20 y 21. Grado 3: 1, 3, 8, 10, 12 y 14. Grado 4: 9, 15. Grado 6: 16.	18
2.4. Ejemplo de una red con topología aleatoria (A) y de una red libre de escala (B).	19
2.5. Deducciones para p (izquierda), q (centro) y p (derecha) para el ejemplo presentado en la sección “ABA frameworks”. El símbolo T representa el valor de verdad verdadero.	28
3.1. Estructura topológica de una red P2P con múltiples capas. Gráfico adap- tado de [MDPY13].	35
3.2. Esquema del algoritmo <i>Routing Indices</i> . Gráfico adaptado de [CGM02]. . .	40
4.1. Red de ejemplo.	48
4.2. Mensajes de consulta y respuesta para el algoritmo 1.0.	49
4.3. Mensajes de consulta y de respuesta para el algoritmo 2.0. El nodo 1 conoce al nodo 5.	50
4.4. Mensajes de consulta y de respuesta para el algoritmo 2.0. El nodo 1 no conoce al nodo 5.	50
4.5. Mensajes de consulta y de respuesta para el algoritmo 2.1.	51

4.6. Mensajes de consulta, respuesta y actualización para el algoritmo 3.0. . . .	52
4.7. Mensajes de actualización para el algoritmo 3.1.	52
4.8. Mensajes de actualización para el algoritmo 3.2.	53
4.9. Mensajes de actualización para el algoritmo 3.3.	54
4.10. Mensajes de consulta, respuesta y actualización para el algoritmo 4.3.3. . .	54
4.11. Red lógica que permite evidenciar el Problema de las Comunidades Cerradas.	58
4.12. Red P2P no estructurada.	60
4.13. Descripción a alto nivel de las componentes de cada nodo n de un framework.	65
4.14. Representación gráfica del envío de mensajes durante el ruteo de consultas en una red P2P (caso de estudio).	67
4.15. Argumentos y ataques involucrados en el árbol de derivación que sopor- ta la decisión “ <i>necesidad_de_explorar</i> ”. Las flechas punteadas representan ataques entre argumentos.	71
5.1. Figura comparativa del número de saltos para encontrar una respuesta ($ N $ $= 1000$).	78
5.2. Figura comparativa del promedio de coeficiente de clustering ($ N = 1000$). .	79
5.3. Figura comparativa del número de mensajes enviados ($ N = 1000$).	80
5.4. Gráfico comparativo del número de mensajes de actualización ($ N = 1000$). .	81
5.5. Distribución de grados para la red física y las redes lógicas resultantes de la primera y de la décima ejecución ($ N = 5000$).	83
5.6. Visualización de la descomposición en k -núcleos para las redes lógicas obte- nidas de la primera (arriba) y de a décima ejecución (abajo) del algoritmo 4.3.3 ($N = 5000$).	84
5.7. Red lógica obtenida a partir de la ejecución del algoritmo 1.0 (ejecución = 10, $ N =1000$).	86
5.8. Red lógica obtenida a partir de la ejecución del algoritmo 4.3.3 (ejecución = 3, $ N =1000$).	86

5.9. Red lógica obtenida a partir de la ejecución del algoritmo 4.3.3 (ejecución = 7, $ N =1000$).	87
5.10. Ejemplo de funcionamiento de la herramienta <i>Proxdd</i> : encontrando argu- mentos a favor de explorar.	88
5.11. Ejemplo de funcionamiento de la herramienta <i>Proxdd</i> : sin argumentos a favor de explorar.	88
5.12. Esquema de la implementación de ArgP2P.	89
5.13. Fragmentos de código de un programa ABA generado por un nodo en <i>ArgP2P</i>	89
5.14. Comparativa del coeficiente de clustering obtenido en la última ejecución de cada una de las simulaciones.	93
5.15. Comparativa de la cantidad de mensajes enviados.	94
5.16. (A) Red lógica obtenida en la décima corrida de la Simulación 1. (B) Red lógica obtenida en la décima corrida de la Simulación 4.	97

Índice de tablas

3.1. Tabla comparativa de las principales características de los algoritmos de ruteo en redes P2P estructuradas.	44
3.2. Aspectos semánticos en algoritmos para redes P2P no estructuradas.	45
3.3. Algoritmos básicos de ruteo en redes P2P no estructuradas.	46
4.1. Resumen comparativo de las principales características de los algoritmos. . .	55
4.2. Conocimiento de los nodos respecto al tópico “matemática”.	66
5.1. Número de artículos asociados a cada tópico.	75
5.2. Tabla comparativa del número de respuestas satisfechas ($ N = 1000$). . . .	77
5.3. Tabla comparativa del número de saltos para encontrar una respuesta ($ N = 1000$).	78
5.4. Tabla comparativa del promedio de coeficiente de clustering ($ N = 1000$). .	79
5.5. Tabla comparativa del número de mensajes enviados ($ N = 1000$).	80
5.6. Tabla comparativa del número de mensajes de actualización ($ N = 1000$). .	81
5.7. Análisis del rendimiento del algoritmo 4.3.3 ($N = 5000$ nodes).	82
5.8. Comparativa del número de veces en la que cada simulación exploró la red. .	92
5.9. Comparativa del coeficiente de clustering promedio en cada ejecución de cada simulación.	93
5.10. Cantidad de saltos en promedio para encontrar una respuesta.	94
5.11. Comparativa de la cantidad de mensajes enviados.	95
5.12. Número total de consultas satisfechas en cada ejecución.	96

Capítulo 1

Introducción

Una red peer-to-peer o red entre pares (red P2P por sus siglas en inglés) es un modelo computacional presente en un gran número de dispositivos, desde teléfonos inteligentes hasta servidores de gran escala. Este modelo permite manejar y distribuir una gran cantidad de datos entre los participantes de la red, balanceando las capacidades de almacenamiento, conectividad y poder computacional entre todos los miembros del sistema. En una red P2P, cada par puede actuar indistintamente como cliente y como servidor, compartiendo información en un entorno distribuido sin ninguna coordinación central. Las redes P2P son utilizadas para compartir datos a gran escala, distribución de contenido y multicast a nivel de aplicación trabajando con un tiempo de espera tolerable para los usuarios [LCP⁺05, Nah04, YXC⁺15].

En estos días, las tecnologías P2P se utilizan en aplicaciones populares tales como *Spotify* y *PopCorn Time*, cuya finalidad es compartir audio y vídeo entre sus usuarios. En entornos educativos, las tecnologías P2P han permitido a las instituciones compartir archivos a nivel mundial, como es el caso del proyecto *LionShare* [Mor07]. Otra aplicación popular distribuida es *Bitcoin* y su alternativas, como *Peercoin* y *Nxt*. *Bitcoin* es un sistema P2P que se utiliza en el ámbito financiero donde las transacciones tienen lugar directamente entre los usuarios, sin ningún intermediario. Estas transacciones son verificadas por los nodos de la red y registradas en un libro público denominado *blockchain*. *Bitcoin* es conocido como la primer moneda digital que funciona en forma descentralizada [Nak, FV17].

Las redes sociales son otra aplicación actual de las redes P2P. Por ejemplo, *Buddy-Cloud* [bud17] es una red social distribuida de código abierto que permite a los desarro-

lladores de software compartir sus aplicaciones, agregando a la vez chats y videos. Otro ejemplo de red social distribuida es *Diaspora** [dia17], cuya política permite la descentralización en el uso de la información. En esta red social cada perfil se almacena en el servidor web personal del usuario, por lo que este tiene control total del contenido que comparte y tiene conocimiento absoluto respecto a dónde se encuentra almacenado y quiénes tienen acceso a él. Otras redes sociales distribuidas incluyen *Friendica* [fri17], *GNU social* [GNU17], *Mastodon* [mas17], *Minds* [min17], *Kune* [kun] y *Twister* [twi17].

Las redes P2P también pueden ser aplicadas a la recuperación de información. Sin embargo, debido a la supremacía de los grandes buscadores, el problema de la recuperación de información en sistemas P2P no ha sido explotado en profundidad. En particular, el desarrollo de servicios tales como buscadores temáticos P2P ofrece gran potencial debido a numerosas ventajas que ofrecen las tecnologías P2P respecto al modelo centralizado a la hora de administrar contenido temático. En el desarrollo de buscadores temáticos P2P en Internet se integra con la computadora personal y cada usuario tiene el potencial de contribuir y colaborar con comunidades específicas, así como con el sistema global. El desarrollo de métodos para la recuperación de información temática en sistemas P2P da lugar un gran número de desafíos de investigación que son el objeto de estudio de la presente Tesis.

1.1. Motivación

El problema de recuperación de información en una red P2P involucra encontrar un conjunto de documentos en la red que sean relevantes a una determinada consulta. Para describir mejor este problema es imprescindible identificar las características más importantes, como se ilustra en la figura 1.1. En una red P2P cada nodo (o par) mantiene una colección de documentos que se encuentra disponible para compartir con los otros pares. Para que los nodos se puedan comunicar entre sí, se necesitan distintas componentes tales como un algoritmo de ruteo, tablas de ruteo, índices y un protocolo para manejar las consultas que arriban a cada uno de los nodos [GCGMP97, DC98, SZTS07]. El algoritmo de ruteo determina cómo un nodo envía los mensajes a través de la red para obtener una respuesta a una determinada consulta. Cuando un nodo recibe un mensaje de consulta, busca documentos relevantes en su colección; si no posee ninguno reenvía el mensaje según lo indique su algoritmo de ruteo. Además, para que todo esto sea posible, se necesita de

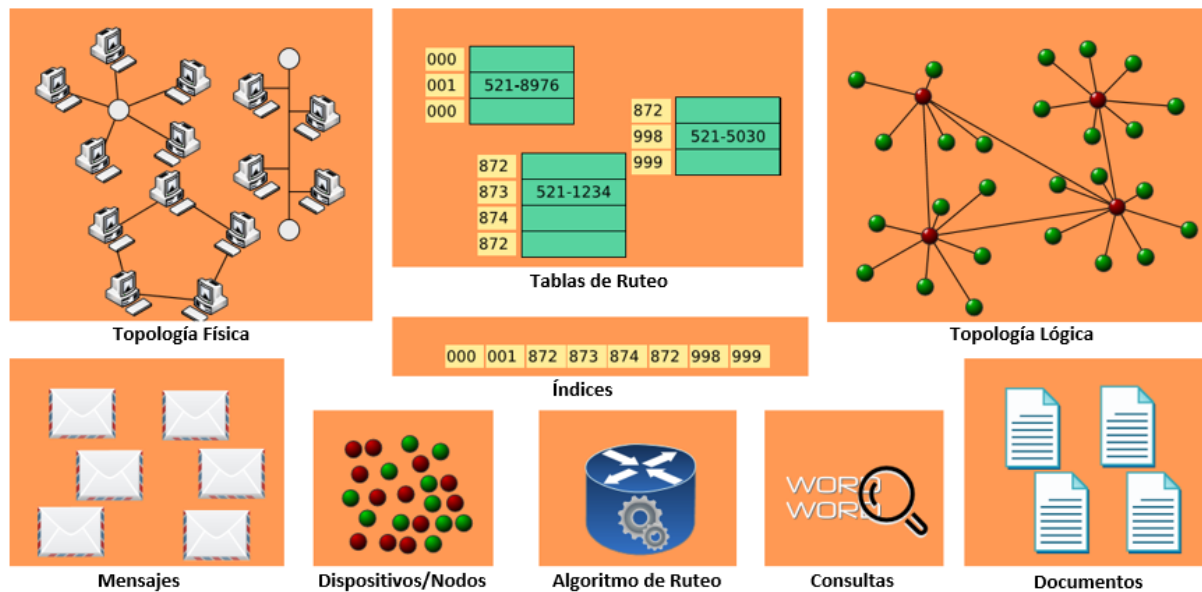


Figura 1.1: Elementos conceptuales que caracterizan una red P2P

una topología subyacente (topología física) y como resultado de la interacción entre los nodos surge una topología lógica.

En la última década se han presentado varias técnicas para recuperación de información en redes P2P [THT12], dando la posibilidad de explotar de mejor manera los recursos por las comunidades presentes en Internet. En este escenario es posible identificar diversos desafíos que hay que afrontar para desarrollar métodos para el acceso a la información distribuida en una amplia variedad de formatos. Las redes P2P ofrecen un gran número de beneficios, entre los que se destacan:

- No requieren ningún tipo de disposición administrativa o financiera.
- Son auto-organizables y adaptables a las necesidades de sus usuarios. Los pares pueden ingresar y salir de la red en forma totalmente libre, sin ningún tipo de regulaciones.
- Pueden manejar una innumerable cantidad de recursos que se encuentran almacenados en millones de dispositivos dispersos a través de Internet.
- Las redes P2P puras son distribuidas y descentralizadas, lo que implica que sean altamente tolerantes a fallos y posean un alto grado de adaptabilidad respecto al balance de la carga.

A través de los años, Internet se ha vuelto restrictiva con las aplicaciones cliente-servidor. Protocolos “poco amigables” y corta fuegos (en inglés, firewalls) son ejemplos de aspectos que restringen y limitan el uso de Internet. Para algunos entendidos, las tecnologías P2P son una forma de retornar a Internet a su concepción inicial, donde se priorizaban los principios cooperativos [Nas03]. El surgimiento de las redes P2P permitió potenciar a casi cualquier grupo de personas con intereses comunes, tales como cultura, política, salud, etc., sorteando muchas de las limitaciones que conlleva el control centralizado.

1.2. Objetivos

Esta Tesis tiene como principal objetivo proponer, investigar y evaluar técnicas de recuperación de información en redes P2P basándose en los intereses de los usuarios. Para abordar este objetivo, se tuvieron en cuenta los siguientes aspectos: contexto temático del usuario, tópicos de interés para los usuarios, tópicos de las consultas, conjunto de documentos que tiene disponible para compartir cada usuario y perfiles de usuarios.

Además de estudiar técnicas de recuperación de información basada en el interés de los usuarios, se realizaron exhaustivas investigaciones y evaluaciones respecto a cómo minimizar la sobrecarga de la red, siendo este uno de los principales objetivos de los algoritmos que se proponen en esta Tesis.

En resumen, los objetivos específicos de esta Tesis son:

- Desarrollar algoritmos colaborativos para la recuperación de información en redes P2P basados en perfiles temáticos de los usuario.

- Presentar un novedoso sistema argumentativo para maximizar el rendimiento de dichos algoritmos.

- Definir estrategias de ruteo y de actualización de datos específicos de cada par con el fin de lograr un aprendizaje local de los participantes.

- Lograr un aprendizaje global de la red a través de la adaptación de su topología lógica definiendo estrategias capaces de explotar el concepto de localidad semántica (a definirse en detalle luego).

- Analizar las ventajas y desventajas de diferentes topologías de red resultantes del aprendizaje global alcanzado.

1.3. Resumen de las técnicas propuestas y contribuciones

En esta sección se presentan las diferentes técnicas propuestas a lo largo de esta Tesis junto con la hipótesis que se pretende investigar.

Algoritmos inteligentes para disminuir el tiempo de respuesta en redes P2P

Las redes P2P son sistemas distribuidos que presentan diversas características que los hacen atractivos, tales como la tolerancia a fallos y la constante disponibilidad de los datos. Existen distintos tipos de redes P2P, entre las cuales se encuentran las redes no estructuradas donde el proceso de búsqueda es dificultoso. Diversos algoritmos se han creado para efectuar estas búsquedas tratando de optimizar el rendimiento de este tipo de sistemas. La hipótesis que guía esta Tesis se basa en el hecho de que si las búsquedas se realizan teniendo en cuenta los intereses de los participantes y el tópico de la consulta surgen *comunidades semánticas* que permiten optimizar el uso de mensajes y la obtención de respuestas a las consultas efectuadas. Es por ello que se han desarrollado una serie de algoritmos que contemplan los siguientes aspectos:

- Análisis de contextos heterogéneos para la búsqueda incremental de información en sistemas distribuidos.
- Uso de perfiles de usuarios donde cada participante del sistema tiene intereses definidos.
- Uso razonable de mensajes para la obtención de respuestas en un tiempo tolerable por el usuario.
- Implementación de mecanismos de aprendizaje para mejorar el sistema de búsqueda de manera incremental.
- Uso de tecnologías adecuadas para la realización de simulaciones.
- Evaluación de los métodos resultantes mediante el uso de métricas tradicionales utilizadas en el área de recuperación de la información y métricas ad-hoc.

Framework ArgP2P

Los algoritmos inteligentes propuestos presentan distintos tipos de comportamientos, entre los cuales se destacan aquellos que maximizan la cooperación entre comunidades que poseen intereses en común. Esta comunicación entre comunidades da lugar a un problema que cierra las comunidades y no permite conocer el resto de las posibilidades que ofrecen los demás participantes de la red. Para sobrellevar este problema se propone en esta Tesis un nuevo framework *ArgP2P* que combina dos comportamientos: uno reactivo (dado por un algoritmo imperativo de aprendizaje incremental) y otro argumentativo (basado en un formalismo para razonamiento argumentativo llamado Assumption-Based Argumentation). El algoritmo imperativo se ejecuta cada vez que un participante procesa un mensaje, mientras que el sistema argumentativo se ejecutará con determinada frecuencia. La idea del sistema argumentativo es que encuentre argumentos que soporten la decisión de explorar la red para poder sobreponerse al problema de la comunicación cerrada entre comunidades y de este modo optimizar los patrones de comunicación. Se desarrollaron distintas simulaciones de *ArgP2P* y de sus componentes por separado para verificar su correctitud y obtener distintos resultados.

Contribuciones

Esta investigación provee las siguientes contribuciones:

1. Desarrollo de nuevos métodos de búsqueda temática para el ruteo de consultas en redes P2P descentralizadas.
2. Desarrollo de nuevos métodos de comunicación eficiente que dan origen a comunidades semánticas.
3. Definición de un marco de evaluación para buscadores temáticos P2P.
4. Definición de una notación formal para el concepto de red P2P junto con una función que permite capturar sus aspectos dinámicos.
5. Desarrollo y evaluación de un framework que condensa los distintos elementos anteriores, y combina un comportamiento reactivo con otro argumentativo para poder resolver eficientemente búsquedas en sistemas P2P descentralizados.

1.4. Organización de la Tesis

Esta Tesis está organizada en los siguientes 6 capítulos:

El presente capítulo (Capítulo 1) presenta una introducción al problema a abordar, motivando su estudio y planteando los objetivos generales y específicos a alcanzar en esta Tesis.

El Capítulo 2 introduce los conceptos necesarios para el desarrollo de esta Tesis. En particular describe los fundamentos de las redes P2P y un marco argumentativo denominado Assumption-Based Argumentation (ABA). Entre estos fundamentos se encuentra la definición de redes P2P, junto con su clasificación y características semánticas y topológicas.

El Capítulo 3 presenta una extensa revisión bibliográfica donde se describen las principales contribuciones en el área de recuperación de información en redes P2P. Además en este capítulo se expone una novedosa taxonomía que clasifica a los algoritmos de ruteo en base al uso de diversos aspectos semánticos.

El Capítulo 4 presenta las contribuciones principales de esta Tesis. En este capítulo se describen algoritmos novedosos para la recuperación de información en redes P2P puras capaces de explotar el concepto de localidad semántica. Además, se presenta un framework argumentativo en el que se combinan algoritmos de ruteo con argumentación, cuyo objetivo es maximizar la rendimiento de las búsquedas temáticas en este tipo de sistemas.

El Capítulo 5 expone un análisis y una comparación de los resultados experimentales obtenidos. Se presenta la evaluación de los algoritmos en forma separada y en conjunto con el sistema argumentativo propuesto en el Capítulo 4.

El Capítulo 6 establece las conclusiones de esta Tesis y señala las líneas de investigación futuras. Para finalizar se incluye un glosario para hacer esta Tesis autocontenida.

Capítulo 2

Fundamentos

En los sistemas distribuidos los participantes intervinientes pueden encontrarse geográficamente dispersos y aún así respetar un protocolo mediante el cual pueden interactuar con el resto de los miembros del sistema. Este protocolo de comunicación es lo que le da la identidad a cada uno de los sistemas distribuidos; en particular tanto un sistema P2P como un sistema cliente-servidor son sistemas que pueden implementarse utilizando tecnologías distribuidas, que poseen objetivos similares pero utilizan un protocolo completamente distinto, lo que hace que ambos sistemas tengan distintos rendimientos. Los sistemas P2P utilizan el poder computacional y ancho de banda de los participantes en lugar de concentrarse en un grupo de servidores. En el modelo cliente-servidor, un nodo o un conjunto limitado de nodos juega el papel de servidor, brindando servicios a otros nodos que actúan como clientes. Por otra parte, en el modelo P2P, cualquier nodo puede ofrecer un servicio a otro. En un sistema P2P puro, los participantes son independientes y tienen las mismas responsabilidades, por lo que un nodo tiene la capacidad de jugar simultáneamente el rol de cliente y de servidor. En términos de rendimiento la ventaja principal del modelo P2P sobre el modelo cliente-servidor es la alta tolerancia a fallos y la flexibilidad de la red respecto a ingreso y egreso de participantes.

En el contexto de redes P2P, es necesario distinguir entre una red *física* y una red *lógica*. La primera está conformada por las conexiones reales que existen entre los dispositivos que forman parte de la red (ya sean conexiones inalámbricas o por cable), mientras que en la segunda la topología emerge de la interacción entre los pares. Muchas veces esta interacción está dada por similitudes semánticas lo que sucede cuando los algoritmos de búsqueda utilizan algún mecanismo que analiza la similitud entre los intereses de

un nodo y el contenido de una determinada consulta. En las redes lógicas que emergen de este tipo de algoritmos se puede evidenciar la existencia de *comunidades semánticas* [CM05, AWMM06].

Una red P2P es un sistema distribuido en sí mismo, pero al mismo tiempo este tipo de redes tiene sus propias clasificaciones que son explicadas en la sección 2.1 junto con los algoritmos que permiten llevar a cabo las búsquedas. En los sistemas P2P cada nodo juega un rol independiente a los demás pero al mismo tiempo tiene los mismos intereses en cuanto a la colaboración para concretar alguna tarea (ej. una búsqueda), naciendo de este modo el concepto de localidad semántica que es explicado en detalle en la sección 2.1.4. Por último, para lograr la colaboración es indispensable implementar mecanismos de búsqueda distribuida bajo los principios de las redes P2P. En su estado puro estas redes tienen una naturaleza no estructurada, lo que hace necesario que los algoritmos de búsqueda tengan que contar con mecanismos inteligentes para rutear las consultas y evitar la propagación ciega, masiva y poco efectiva de las mismas. En la propuesta que es el tema central de esta Tesis, estos mecanismos inteligentes incluyen una estrategia orientada a explotar el concepto de localidad semántica y la aplicación de un sistema argumentativo formalizado bajo el framework Assumption-Based Argumentation(ABA) que es presentado en la sección 2.2. Para finalizar el capítulo se brinda en resumen del mismo en la sección 2.3.

2.1. Características y clasificación de las redes P2P

Una red P2P es, en su forma más pura, un sistema distribuido en el cual cada par (nodo) se comunica con otros sin intervención de hosts localizados [Sch01]. En la práctica la mayoría de las computadoras que son parte de una red P2P corresponden a usuarios finales, tales como una computadora doméstica o de oficina [KR02]. Por este motivo, una red P2P consiste fundamentalmente de un conjunto de computadoras personales, aunque cada una puede contar con diferentes capacidades de procesamiento, almacenamiento y velocidad en el acceso a la red. Este tipo de redes poseen propiedades que las hacen atractivas para tareas tales como compartir archivos, streaming y búsqueda distribuida.

Las redes P2P generalmente no cuentan con un directorio centralizado o punto de control, y es por ello que no tienen un punto central de falla. Esta propiedad hace que las redes sean *auto-organizadas*, lo que significa que automáticamente se adaptan al ingreso de un nuevo nodo, así como también al egreso o la falla de alguno de los mismos. Para que

los nodos de la red puedan comunicarse se usa un lenguaje común o protocolo *simétrico*, lo que significa que cuando dos nodos están conectados, tienen la misma capacidad de comunicación. Esta simetría hace que la red sea escalable, en el sentido de que cada vez que un nodo se une a la misma, agrega más poder computacional al total de la red [BMR03, RM06, BMNV13]. Existen muchos desafíos en las redes P2P [DGMV03], entre los que se incluyen cómo administrar adecuadamente los recursos y cómo garantizar una adecuada calidad del servicio sin perder de vista la robustez y la disponibilidad de los datos. Revisiones de diferentes frameworks para redes P2P y sus aplicaciones pueden encontrarse en [ATS04, Pas12, Mal15, ZI16].

En la sub-sección 2.1.1 se presenta una clasificación tradicional de las redes P2P. La sub-sección 2.1.2 introduce el concepto de *aspecto semántico* en el contexto de una red P2P, una característica presente en muchos de los algoritmos de recuperación de información más avanzados. Estos algoritmos pueden clasificarse según el tipo de mecanismo que utilizan para guiar la búsqueda; esta taxonomía es presentada en la sub-sección 2.1.3

2.1.1. Clasificación

La forma tradicional de clasificar a las redes P2P se basa en su *grado de descentralización* y en su *estructura*:

- **Centralizadas:** Este tipo de redes P2P posee una estructura monolítica, con un único servidor que habilita las transacciones entre nodos y mantiene un seguimiento respecto a dónde se encuentra almacenado el contenido. Un conocido ejemplo de una red P2P centralizada es Napster [Nap], una red con un directorio central popularmente conocido. Este sistema fue extremadamente exitoso antes de sus dificultades legales. En general el problema con estas redes centralizadas es que son poco escalables y su directorio centralizado se vuelve un único punto de falla.
- **Descentralizado y no estructurado:** En este tipo de redes no existe un directorio central ni ningún otro punto de control respecto a la topología o el lugar físico en el que se encuentra ubicado el contenido. Gnutella [Rip01] es un ejemplo de este tipo de arquitectura (una red formada por nodos que ingresan y egresan de la red casi en forma constante). Hay muchos factores que motivaron al uso de redes descentralizadas tales como el control de la privacidad, la disponibilidad del contenido, la

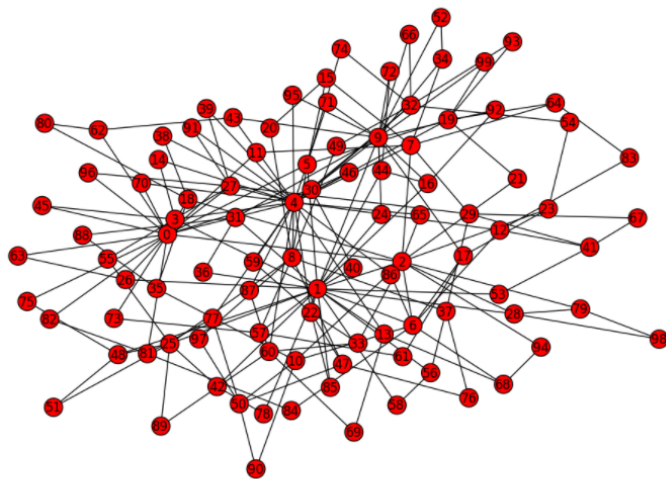


Figura 2.1: Ejemplo de red P2P no estructurada.

escalabilidad, la seguridad y la confiabilidad [QMB⁺16]. Un aspecto controvertido de este tipo de redes es la forma en la que se localizan los contenidos. La forma tradicional es que cada nodo envíe la consulta a todos los vecinos. Este método llamado *Inundación* (del inglés Flooding) es extremadamente ineficiente, generando grandes cargas en los nodos de la red.

- **Híbrido o Descentralizado pero estructurado:** Estos sistemas no tienen un directorio centralizado, y por lo tanto son considerados redes descentralizadas. Más allá de eso, poseen cierta estructura en su topología (ej. árbol, anillo) que garantiza encontrar respuesta a las consultas en un determinado tiempo de ejecución (ej. logarítmico). FreeNet [CSWH01, FRE11] es un ejemplo de este tipo de sistemas en los que hay un creciente interés [RFH⁺01, SMK⁺01, RD01, ZHS⁺04, HWS13].

Los métodos de búsqueda dependen fuertemente de la estructura y del grado de centralización de la red. Una red puramente descentralizada, tal como la que se presenta en la figura 2.1, no sigue una estructura en la organización de sus nodos. Por esta falta de estructura es que la búsqueda es relativamente ineficiente comparada con las búsquedas en sistemas estructurados o híbridos. En un sistema descentralizado, una única consulta puede generar un aumento masivo de tránsito aún en una red de tamaño moderado [RM06]. Para solucionar este problema, se desarrollaron varios métodos que mejoran el algoritmo

básico de Inundación o *Flooding*, entre los que se encuentran algoritmos estilo “*random walk*”, que envían las consultas a nodos vecinos seleccionados aleatoriamente [LCC⁺02, KGZY02a, YL13, BFR14, LYHC14], algoritmos que dirigen las consultas a nodos potencialmente útiles [ALPH01, ZYKG04, WW13, SCG⁺16], y algoritmos que agrupan a los nodos en forma de clusters según su contenido [CGM05, UAO13, BGSWW14] o intereses [SMZ03, MDPY13, QYL14].

2.1.2. Aspectos semánticos

Una posible solución para mejorar la sobrecarga en las comunicaciones y la escalabilidad en una red P2P no estructurada es enviar las consultas a un grupo de pares que pueden tener contenido potencialmente útil. La selección de estos nodos potencialmente útiles está basada en el historial de los nodos o en la similitud semántica respecto a los intereses del nodo demandante o respecto al contenido de la consulta en sí [CGM02, SMZ03, VKMvO04, KJ07, LC07, LWAM09, THI⁺10]. Para identificar estos pares potencialmente útiles los algoritmos utilizan los aspectos semánticos en diferentes formas.

Un *aspecto semántico*, en el contexto de una red P2P, es una característica o grupo de características que permiten reconocer la semántica de los datos almacenados en un determinado nodo de la red. Los aspectos semánticos son utilizados por los algoritmos de búsqueda para ayudar a predecir cuáles pares pueden tener contenidos lo suficientemente útiles como para satisfacer la consulta. La información respecto a tópicos, la experiencia pasada y el estado de los nodos son ejemplos de aspectos semánticos.

En los algoritmos en los que se utiliza la información respecto a tópicos cada nodo almacena los perfiles. Un perfil de un nodo es información que un par mantiene para describir la información que contiene su vecino. Analizando estos perfiles, los nodos incrementan la probabilidad de obtener una respuesta eligiendo a los candidatos más apropiados. En general, aquellos algoritmos que utilizan información respecto a tópicos para la implementación de su mecanismo de búsqueda, resultan en la formación espontánea de comunidades semánticas que surgen a partir de la interacción entre los nodos [AWMM06].

El concepto de “localidad semántica” [YZ04] es uno de los conceptos centrales asociado con el uso de similitud de tópicos en el contexto de búsqueda en redes P2P. Tradicionalmente, la noción de localidad semántica ha sido utilizada para referirse a la capacidad

de almacenar información semánticamente similar entre pares que tienden a interactuar entre sí. Por este motivo se relaciona fuertemente con el hecho de encontrar e indexar contenidos semánticamente similares en la Web [SSDN02, ZWH03, SP04, LKG13]. Con la ayuda de la información de la localidad semántica, una red P2P no estructurada ofrece un mecanismo de búsqueda más informado e inteligente, mitigando de este modo la complejidad del proceso en sí [NLMC13, NMC15].

Otro acercamiento es almacenar la experiencia o historia pasada respecto a la interacción de un determinado par con sus vecinos (directos o no directos). En este enfoque no se guarda el perfil de un nodo sino que se mantiene un registro de información de ruteo tal como el número de hits y la disponibilidad. Esta información también es usada para la selección de nodos potencialmente útiles para enviar una determinada consulta [DZGL05]. El principal problema con esta propuesta es que solamente beneficia a aquellos pares que almacenan mayor cantidad y variedad de contenidos, relegando a los pares que quizás tengan menor cantidad de contenido pero que aún así ofrecen material relevante respecto a tópicos específicos.

Una gran cantidad de algoritmos usan heurísticas basadas en el estado de los nodos o en su rendimiento pasado para seleccionar nodos candidatos. La información heurística comúnmente utilizada se basa en el análisis de latencia o en el tiempo de respuesta [ZHF10]. Claramente, para realizar este tipo de búsquedas, hay que almacenar información específica para poder computar la información heurística en cuestión.

Para resumir, el uso de aspectos semánticos permite seleccionar nodos potencialmente útiles con el propósito de implementar estrategias de búsqueda más avanzadas.

2.1.3. Clasificación de los algoritmos

En redes P2P no estructuradas, los algoritmos de búsqueda pueden clasificarse según la forma en que explotan la información semántica en *orientados al contenido* u *orientados a la consulta*. El criterio que utiliza el algoritmo para la toma de decisiones respecto a cómo rutear una dada consulta es fundamental para determinar la clasificación del algoritmo respecto a su semántica [AY14].

- **Orientados al contenido:** Estos algoritmos usan meta-datos respecto a cada par con el objetivo de construir un índice con información global. Este índice le brinda

a cada par una visión aproximada del contenido de la red y de los perfiles de otros nodos. De este modo los nodos pueden rutear en forma eficiente sus consultas, mejorando significativamente el resultado de las búsquedas. El uso de perfiles de los participantes es el aspecto semántico más usado en los algoritmos orientados al contenido [CGM02, KXZ05].

- **Orientados a la consulta:** Este tipo de algoritmos explota la información pasada para rutear las consultas futuras. La experiencia pasada basada en la cantidad de hits es el aspecto semántico más utilizado en algoritmos orientados a la consulta [TR03, LW06].

Los algoritmos orientados al contenido generan una gran cantidad de mensajes para poder construir sus índices. Por otra parte, los algoritmos orientados a la consulta son más ventajosos en el sentido que no sobrecargan en forma excesiva la red para crear sus índices. En años recientes, se han propuesto enfoques orientados al contenido en el contexto de “redes centradas en el contenido” [CHC⁺11, YGS15]. En estas redes, un objeto de dato es recuperado en base a su contenido en lugar de recuperarlo basándose en una dirección IP del nodo en el que reside.

2.1.4. Características topológicas de una red lógica

Como se mencionó anteriormente, una posible solución para mejorar el rendimiento y la escalabilidad en grandes sistemas P2P no estructurados es enviar las consultas hacia un grupo de pares que sean potencialmente útiles. La selección de los nodos potencialmente útiles típicamente se basa en su rendimiento pasado o en la similitud semántica respecto a la consulta original [BCAA04, VKMvO04]. Una *comunidad semántica*, en el contexto de una red P2P, es un grupo de nodos que comparten información relacionada. Un concepto fuertemente asociado al de comunidad semántica es el de *localidad semántica* [YZ04]. Como se mencionó anteriormente, este concepto ha sido utilizado para referirse a la habilidad de almacenar información semánticamente similar entre pares que tienden a interactuar entre sí. También se ha definido el concepto de localidad semántica como “una categorización semántica lógica de un grupo de pares que comparten datos en común” [Sha06]. El rendimiento de los algoritmos de búsqueda en redes P2P puede mejorarse aplicando mecanismos que exploten la localidad semántica de los datos. Este tipo de algoritmos

permite predecir para cada par, qué otros tienen información útil para responder a una consulta, en la menor cantidad de saltos, sin sobrecargar el sistema [THI⁺10]. Con este tipo de enfoques se hace posible que una consulta sea eficientemente propagada en la red reenviando el mensaje solamente a nodos relevantes, sugiriendo que la búsqueda colaborativa y distribuida se beneficia del contexto del usuario y de la existencia de comunidades semánticas.

Para analizar la topología de una red P2P es posible recurrir a determinados conceptos tales como coeficiente de clustering y topología de mundo pequeño, los que son introducidos a continuación.

Coeficiente de clustering

El coeficiente de clustering local cuantifica la tendencia de los nodos a agruparse en una red inmediata a un único nodo [WS98]. Consideremos un grafo no dirigido $G = (V, E)$ para representar la red, donde V es el conjunto de nodos y E el conjunto de arcos. Utiliza la notación e_{ij} para representar un arco entre el nodo v_i y v_j . Para un nodo v_i , su vecindario N_i está definido como el conjunto de nodos v_j inmediatamente conectados a v_i , esto es:

$$N_i =_{def} \{v_j \mid e_{ij} \in E \vee e_{ji} \in E\}.$$

El coeficiente de clustering local está basado en la densidad de la red o densidad local [Bur92, EB05]. Para cada nodo v_i , se mide como la fracción del número de arcos que conectan los vecinos de v_i sobre el número total de posibles arcos entre los vecinos de v_i . Sea k_i el número de vecinos de un nodo v_i , esto es, $|N_i|$. Si un nodo v_i tiene k_i vecinos entonces hay a lo sumo $k_i(k_i - 1)/2$ arcos entre los nodos en el vecindario (si éste está totalmente conectado). Por lo tanto, el coeficiente de clustering local puede calcularse en un grafo no dirigido como:

$$C_i = \frac{2|e_{jk} \in E : v_j, v_k \in N_i|}{k_i(k_i - 1)}.$$

Para calcular el coeficiente de clustering de toda la red, se promedia el coeficiente de clustering local de cada nodo [WS98]. Sea n el número de vértices o nodos de la red, esto es $|V|$. El promedio del coeficiente de clustering se calcula como:

$$C_{average} = \frac{1}{n} \sum_{i=1}^n C_i.$$

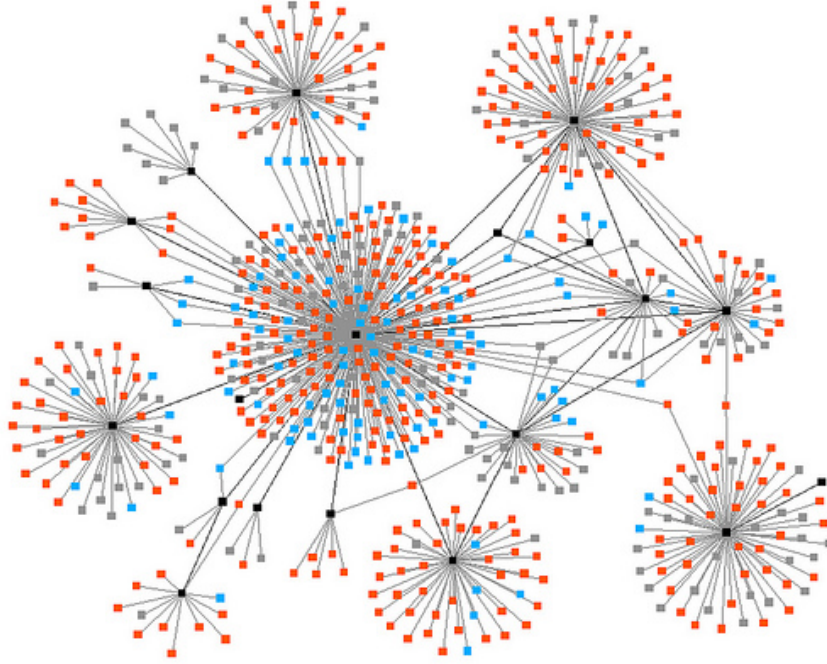


Figura 2.2: Ejemplo de una red con topología de mundo pequeño.

Topología de mundo pequeño

Se considera que un grafo tiene una topología de mundo pequeño si se verifica simultáneamente que sus enlaces son globalmente escasos, el coeficiente de clustering $C_{average}$ es superior al de un grafo aleatorio y la distancia entre cualquier par de nodos de la red es de varios órdenes de magnitud más pequeña que el tamaño de la red [WS98]. Es decir, la red está lejos de ser totalmente conectada y aún así su coeficiente de clustering es relativamente alto y el diámetro de la red (distancia máxima existente entre dos nodos) es pequeño. En la figura 2.2 se muestra un ejemplo de una red con topología de mundo pequeño.

Esta noción está directamente relacionada con el conocimiento global de una red y es utilizada en el marco de esta Tesis para comparar la capacidad de los algoritmos para difundir la información asociada con los nodos en la red. Cuando el conocimiento global de la red (de cada nodo respecto a los demás) es escaso o insuficiente, el valor de $C_{average}$ es pequeño. Sin embargo, a medida que el conocimiento se incrementa también lo hace el valor de $C_{average}$. Como consecuencia de esto último, sólo se necesitarán unos pocos saltos

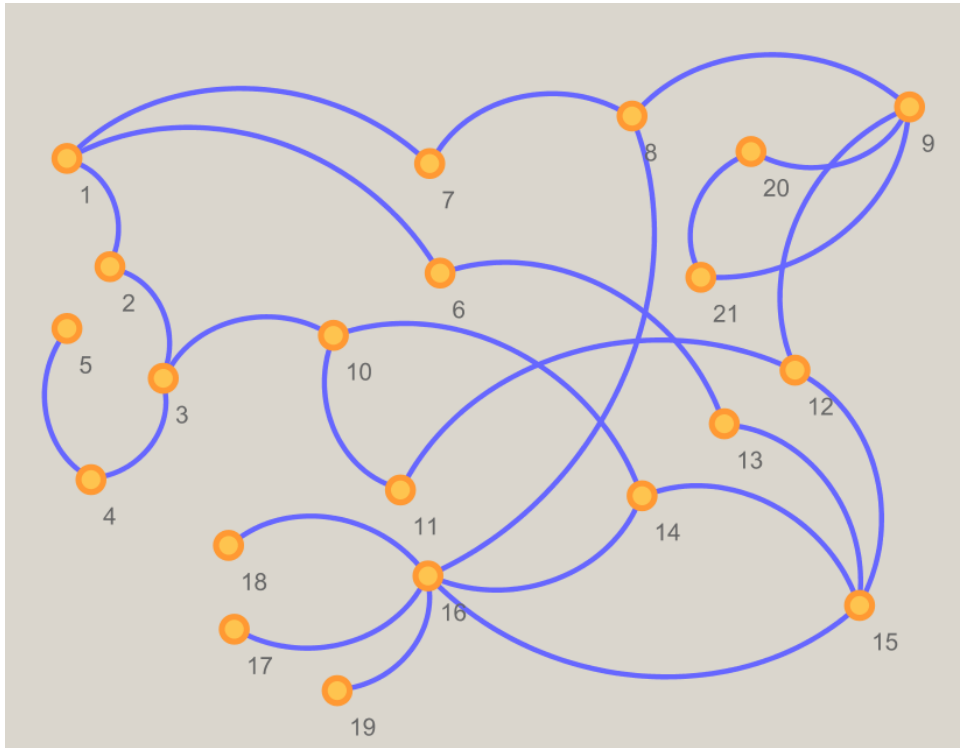


Figura 2.3: Ejemplos del grado de un nodo. Grado 1: 5, 17, 18 y 19. Grado 2: 2, 4, 6, 7, 11, 13, 20 y 21. Grado 3: 1, 3, 8, 10, 12 y 14. Grado 4: 9, 15. Grado 6: 16.

para enviar un mensaje de un nodo hacia otro, indicando que la longitud del camino que conecta ambos nodos es pequeña.

Grado de los Nodos y Redes Libres de Escala

Como se ha mencionado anteriormente, una red P2P puede considerarse como un grafo no dirigido. Una propiedad importante de los grafos es el *grado de sus nodos*. El *grado de un nodo* en particular es un entero no negativo que indica la cantidad de arcos que inciden en el nodo en cuestión. Como se puede observar en la figura 2.3 en una misma red los nodos pueden presentar distintos grados, en este caso los nodos 5, 17, 18 y 19 tienen grado 1, los nodos 2, 4, 6, 7, 11, 13, 20 y 21 tienen grado 2, los nodos 1, 3, 8, 10, 12 y 14 tienen grado 4 y el nodo 16 tiene grado 6.

Mediante el cálculo de los grados de los nodos de una red cualquiera se pueden calcular distintas distribuciones. En particular, si respeta una distribución de potencias se puede decir que dicha red es una red *libre de escala* [BB03]. En algunos casos las topologías libres

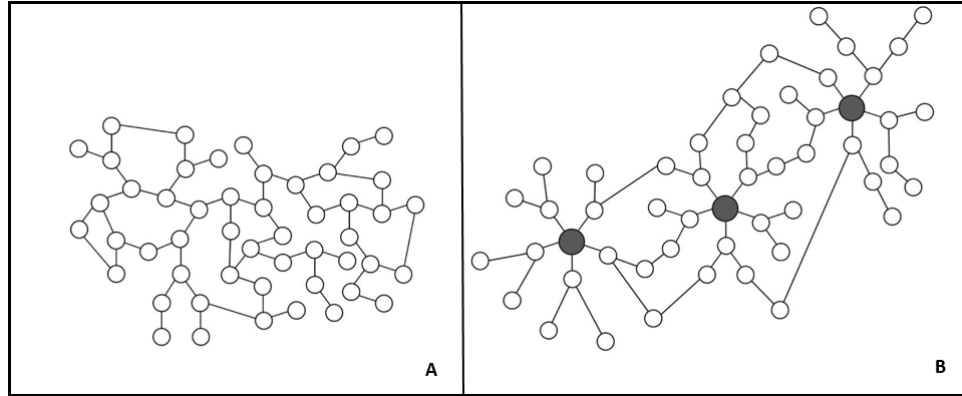


Figura 2.4: Ejemplo de una red con topología aleatoria (A) y de una red libre de escala (B).

de escala emergen por un determinado comportamiento de sus participantes como es el caso de algunas redes sociales tales como las relaciones actores-películas [FGT96] y co-autores [Kre04]. En otros casos este tipo de topologías puede observarse en la organización de sistemas de transporte (tales como las redes de aeropuertos), en sistemas financieros (tales como las redes de comercio) e incluso en sistemas biológicos (tales como las redes de interacción de proteínas). Sin duda alguna la red que respeta esta topología y es la más popular es Internet y la World-Wide-Web. En la figura 2.4 se puede observar gráficamente la diferencia entre una red aleatoria (figura 2.4 A) y una red libre de escala (figura 2.4 B). Nótese que en esta última los nodos hubs se encuentran en color gris.

2.2. Razonamiento no-monótono y argumentación rebatible

La lógica matemática fue desarrollada para formalizar hechos precisos y razonamiento válido (o correcto). Cuando un agente razona utilizando lógica matemática pura, se asume que todos los hechos relevantes son conocidos a priori. Sin embargo, cuando lidiamos con el mundo real, esta hipótesis no es cierta ya que usualmente no contamos con todos los hechos de la realidad. Además, la lógica matemática clásica es de naturaleza monótona; esto quiere decir que cuando agregamos nuevo conocimiento no es posible descartar conocimiento adquirido previamente.

En la matemática el conocimiento se representa con lógica de primer orden (FOL), lo cual es un poderoso formalismo de representación de conocimiento. Sin embargo, la FOL tiene severas restricciones para las aplicaciones de Inteligencia Artificial (IA). Estas restricciones incluyen la monotonidad y su limitada habilidad para representar información incierta o incompleta. Además, se presentan problemas cuando la información que se posee es contradictoria. Así, el razonamiento de sentido común es la habilidad de razonar en base a información incompleta y cambiar nuestros pensamientos cuando aparece nueva información (razonamiento no-monótono).

En su tratamiento clásico dentro de la filosofía, el estudio de la argumentación puede considerarse informalmente con la forma en que se proponen, discuten y resuelven las aserciones en un contexto donde se encuentran distintas opiniones. Así, las investigaciones filosóficas de la argumentación, desde Aristóteles hasta nuestros días, han abordado temas tales como: los mecanismos por los cuales se puede distinguir la argumentación “legítima” en apoyo de una demanda de la argumentación “defectuosa”; análisis de las estructuras típicas que constituyen componentes en el desarrollo de la argumentación; los procesos por los cuales los participantes del debate pueden avanzar en sus respectivas posiciones y socavar posturas y argumentos encontrados, etc. y los contextos en los que se deciden estas cuestiones. La importancia de tales teorías filosóficas para el llamado razonamiento cotidiano tiene una larga y distinguida historia en IA, y las contribuciones de los análisis filosóficos contemporáneos siguen desempeñando un papel importante en la evolución de la explotación computacional efectiva de la argumentación.

En las últimas décadas la argumentación se ha convertido en un tema central de estudio en el área de la IA. Existen distintos tipos de tendencias, desarrollos y aplicaciones que cubren todo un rango de diversos tópicos relacionados a la teoría y aplicación de la argumentación. En [BCD07] se puede encontrar una introducción a las principales contribuciones históricas en el área de la argumentación en IA y una discusión respecto a una serie de cuestiones que han surgido en los últimos años que conllevaron a una ampliación significativa de las áreas en las que se utilizan métodos basados en la argumentación.

En la década del 80, la argumentación emergió en el área de Inteligencia Artificial como un método poderoso para representar conocimiento de sentido común y combinarlo con razonamiento cualitativo para manejo de información potencialmente inconsistente. La argumentación es una manera de entender y reconciliar diferencias y similitudes de varios formalismos existentes (por ejemplo, lógica no monótona y razonamiento por defecto)

que se estudian dentro del área de IA [SR09]. Esta línea de investigación ha llevado al desarrollo de los diferentes sistemas de razonamiento basados en argumentación; sobre todo la argumentación abstracta (AA) [Dun95] y la argumentación basada en suposiciones (ABA son sus siglas en inglés, *Assumption-Based Argumentation*) [BTK93]. Más allá de esto, comenzando con Pollock [Pol87], la argumentación fue identificada como una manera de entender el razonamiento rebatible tal como es estudiado en filosofía, dando lugar al surgimiento de sistemas tales como *DeLP* [GS03] y *ASPIC+* [MP13].

Como se mencionó anteriormente, uno de los desafíos en el campo de la Inteligencia Artificial es modelar el razonamiento de sentido común, el cual suele ser inconsistente e incompleto. Los modelos lógicos del razonamiento de sentido común demandan la formalización de principios y criterios para caracterizar patrones válidos de inferencia. La lógica clásica ha probado ser inadecuada para este fin, ya que se comporta en forma monótona y no permite lidiar con inconsistencias a nivel de objeto [Rei80].

Cuando una regla apoyando una conclusión puede ser derrotada por nueva información, se dice que tal razonamiento es *rebatible*. Cuando tales razones rebatibles o reglas son encadenadas para llegar a una conclusión, tenemos *argumentos* en lugar de pruebas. Los argumentos pueden competir, derrotándose entre sí, de tal manera que un proceso de argumentación es el resultado natural de la búsqueda de argumentos. La adjudicación de argumentos que compiten entre sí debe ser realizada comparando argumentos para determinar qué creencias están últimamente aceptadas como *garantizadas* o *justificadas*.

La argumentación provee una perspectiva diferente al razonamiento no-monótono y al razonamiento rebatible, en el cual una afirmación es aceptada o rechazada en base a los argumentos en favor o en contra de ella, y en base al hecho de si dichos argumentos son atacados y derrotados por otros [CRL00]. Esta visión es sustentada por la *argumentación rebatible*.

Los frameworks de argumentación tienen generalmente los siguientes elementos [CML00]: un lenguaje lógico subyacente, un concepto de argumento, un concepto de conflicto entre argumentos, una noción de derrota entre argumentos y una noción de cuándo un argumento es aceptable. A continuación se describen cada uno de estos elementos:

- **Lenguaje lógico subyacente:** El lenguaje lógico subyacente será el de la lógica formal de primer orden donde están definidas una o más relaciones de consecuencia

monótonas que establecen la base para derivar argumentos. En general, muchos sistemas de argumentación involucran una base de conocimiento $K = (\Pi, \Delta)$ que provee el conocimiento del dominio para un agente formalizado en un lenguaje de primer orden L . Este conocimiento de dominio involucra usualmente un conjunto Π de reglas estrictas y hechos y un conjunto Δ de reglas rebatibles.

- **Argumento:** Un argumento es una prueba rebatible obtenida a partir de la base de conocimiento K por medio de la aplicación de reglas de inferencia (quizás rebatibles) asociadas con el lenguaje lógico subyacente L .
- **Conflicto entre argumentos:** Dados dos argumentos A y B , el *conflicto* (o ataque) entre argumentos surge siempre que A y B no puedan ser aceptados simultáneamente, típicamente debido a algún tipo de contradicción lógica.
- **Derrota entre argumentos:** Como el lenguaje lógico subyacente es monótono, el agregado de nueva información no invalida argumentos existentes ni conclusiones derivadas previamente; por lo tanto, en una base de conocimiento pueden coexistir argumentos en conflicto. El carácter no-monótono de la argumentación surge del hecho de que algunos argumentos serán preferidos sobre otros y el formalismo debería tener medios para decidir cuál de dichos argumentos es aceptable.

La noción de derrota está usualmente basada en alguna medida comparativa para los argumentos y se utiliza algún criterio basado en esta medida para decidir la identidad del argumento ganador. Una manera es asignar prioridad a las reglas de un sistema basado en reglas, otra forma son los criterios de especificidad y el de directitud.

Formalmente, muchos sistemas de argumentación proveen un criterio de preferencia que define un orden parcial entre argumentos, permitiendo determinar cuándo A debe preferirse sobre B . Esto define una relación de *derrota*. Dado el conjunto $Args$ de argumentos obtenidos de una base de conocimiento K , vale que la relación de ataque $attacks \subseteq Args \times Args$. Cuando el argumento A se prefiere sobre un argumento B , se dice que A derrota a B .

- *Argumento aceptable:* El objetivo de un sistema argumentativo es el de determinar qué afirmaciones y qué argumentos son aceptables. La noción de aceptabilidad varía de formalismo en formalismo, pero intuitivamente la noción se corresponde con

que un argumento no será aceptable si es derrotado por algún argumento aunque sea capaz de derrotar a un argumento en conflicto con él. Para determinar si un argumento dado A es *aceptable*, se lleva a cabo un proceso dialéctico, en el cual se tienen en cuenta derrotadores para A , derrotadores para estos derrotadores y así sucesivamente. Por ello, para este proceso se deben tener en cuenta a todos los argumentos a favor y en contra de una afirmación dada antes de tomar una decisión.

2.2.1. Argumentación basada en suposiciones

ABA está estrechamente relacionado con el concepto de Argumentación Abstracta (AA), en el sentido de que es un ejemplo de ello pero al mismo tiempo admite la AA como instancia [Ton12]. Sin embargo, la AA toma un conjunto de argumentos dado y una relación de ataque entre este conjunto de argumentos y se centra en determinar qué conjuntos de argumentos pueden ser considerados como “ganadores” (según diferentes semánticas). Pero en ABA, los argumentos y los ataques son derivados de reglas dadas en un sistema deductivo, las suposiciones y sus contras. Pero al mismo tiempo ABA también es abstracto en el sentido que, tal como en AA, no se compromete a ninguna instancia específica y admite, en su lugar, varias instancias (incluyendo AA). Sin embargo, las componentes de ABA (reglas, suposiciones y contras) están en un nivel de abstracción más bajo que los de AA (argumentos y ataques). ABA se encuentra equipado con diferentes semánticas para determinar los conjuntos de argumentos “ganadores”. Estos conjuntos ganadores pueden ser entonces determinados de una manera directa a partir de los conjuntos de supuestos ganadores (el nivel de suposición y argumento son totalmente equivalentes en ABA). Manipulando directamente las suposiciones (y los argumentos sólo en forma indirecta) en lugar de los argumentos, es ventajoso en términos computacionales ya que esto permite, en forma implícita, explotar las superposiciones entre argumentos y evitar recalcularlos, así como terminar, incluso en presencia de bucles, para determinar si una conclusión puede ser apoyada por un conjunto de argumentos ganadores. ABA cuenta con un catálogo de técnicas computacionales [DKT06, DMT07, Ton13] que tienen sus bases en la programación lógica. Estas técnicas pueden ser utilizadas para determinar cuándo las conclusiones dadas pueden ser soportadas por un conjunto ganador de argumentos. Se conocen como “disputas”, que se aplican junto a técnicas de filtrado para evitar recalcular reglas ya computadas. Estas disputas pueden ser vistas como juegos (ficticios) entre el proponente y los jugadores oponentes, y dependen de estructuras de datos para

representar el estado del juego en un momento dado. La complejidad computacional de ABA con distintas semánticas e instancias ha sido estudiada en [DNT02, Dun09].

Tal como se mencionó anteriormente, ABA es un framework de propósito general que puede instanciarse para soportar varias aplicaciones y frameworks especializados, incluyendo frameworks de razonamiento por defecto (en inglés, default reasoning) [BTK93], problemas de razonamiento legal [DKT09], teoría de juegos [Dun95], razonamiento práctico y teoría de decisiones [Ton08, MTSD08, B⁺08, GS04]. Este framework argumentativo posee numerosos campos de aplicación, entre los más destacadas se encuentran los siguientes:

- Uso de la argumentación para resolver problemas de clasificación;
- Justificación de respuestas;
- Aprendizaje por refuerzo basado en argumentación (RoboCup-Soccer);
- Argumentación para la toma de decisiones con preferencias;
- Resolución de conflictos entre agentes;
- Uso de la argumentación para la toma de decisiones en medicina;
- Combinación de estadísticas y argumentación para el cómputo de la confiabilidad;
- Resolución de conflictos;
- Toma de decisiones.

Este estilo computacional, que tiene su origen en programación lógica, tiene numerosas ventajas sobre otros mecanismos computacionales para argumentación. Las ventajas se deben principalmente al fino nivel de granularidad proporcionada por la forma en la que se construyen los argumentos y en la que se determina su aceptabilidad. Por varios motivos, incluido el mencionado anteriormente, ABA se ha vuelto particularmente aplicable en el contexto de problemas reales (por ejemplo, grid computing [TGK⁺08]). Es por todos estos motivos que se ha seleccionado ABA para que proporcione el marco argumentativo necesario para el desarrollo de la contribución principal de esta Tesis: *ArgP2P*. En este caso en particular la noción de “suposiciones” es la adecuada para el manejo de la información

no corroborada. Por ejemplo, un nodo de la red puede suponer que otro es un buen candidato para responder a una dada consulta, pero no lo puede asegurar.

En lo que resta de esta sección se presentan las principales definiciones y conceptos de ABA. Para obtener una descripción completa y detallada de ABA junto con sus aplicaciones se remite al lector a [Ton14].

ABA framework

Un framework ABA es una terna $\langle L, R, A \rangle$ donde:

- $\langle L, R \rangle$ es un sistema deductivo, con L el *lenguaje* y R un conjunto de *reglas*, que se asumen de la forma $\sigma_0 \leftarrow \sigma_1, \dots, \sigma_m$ ($m \geq 0$) con $\sigma_i \in L$ ($i = 0, \dots, m$); σ_0 se conoce como la *cabeza* y $\sigma_1, \dots, \sigma_m$ como el *cuerpo* de la regla $\sigma_0 \leftarrow \sigma_1, \dots, \sigma_m$;
- $A \subseteq L$ es un conjunto (no vacío), conocido como *suposiciones*;
- $\bar{\cdot}$ es un mapeo total de A en L ; \bar{a} es conocido como el *contrario* de a .

Ataques y Argumentos

En ABA, los *argumentos* son deducciones obtenidas a partir de las reglas y soportadas por un conjunto de suposiciones, mientras que los ataques son dirigidos a las suposiciones que soportan a los argumentos:

- *Un argumento para $\sigma \in L$ soportado por $A \subseteq A$ ($A \vdash \sigma$) es una deducción para σ soportado por A (y algún $R \subseteq R$).*
- *Un argumento $A_1 \vdash \sigma_1$ ataca a otro argumento $A_2 \vdash \sigma_2$ si y sólo si σ_1 es el opuesto de una de las suposiciones en A_2 .*

En ABA, los ataques entre argumentos corresponden a ataques entre conjuntos de suposiciones, donde un conjunto de suposiciones A ataca a un conjunto de suposiciones A' si y sólo si un argumento soportado por un subconjunto de A ataca a un argumento soportado por un subconjunto de A' . En general:

- Si un argumento α ataca otro argumento α' entonces el conjunto de suposiciones que soporta a α ataca al conjunto de suposiciones que soportan α' ;
- si un conjunto de suposiciones A ataca a otro conjunto de suposiciones A' entonces algún argumento soportado por un subconjunto de A ataca a algún argumento soportado por un subconjunto de A' .

Semánticas

Las semánticas argumentativas ofrecen métodos declarativos para determinar conjuntos de argumentos “aceptables”, llamados conjuntos de argumentos “ganadores” desde un punto de vista dialéctico. En ABA, estas semánticas se pueden usar equivalentemente para determinar conjuntos de suposiciones aceptables o ganadoras.

Formalmente, un conjunto de argumentos A es:

- *Admisible* si y sólo si no se ataca a sí mismo y ataca a todos los argumentos que lo atacan;
- *Preferido* si y sólo si es maximal.
- *Escépticamente preferido* si y sólo si es la intersección de todos los conjuntos de argumentos preferidos;
- *Completo* si y sólo si es admisible y contiene todos los argumentos que lo defienden, donde A defiende a α si y sólo si A ataca a todos los argumentos que atacan a α ;
- *Grounded* si es minimal;
- *Ideal* si y sólo si es maximal, admisible y contenido en todos los conjuntos de argumentos preferidos;
- *Estable* si y sólo si no se ataca a sí mismo y ataca a todos los argumentos que no contiene.

ABA también se encuentra equipado con la noción de “aceptabilidad” de conjuntos de suposiciones, reflejando las nociones de “aceptabilidad” de conjuntos de argumentos pero con la noción de ataque entre (conjuntos de) argumentos reemplazada por la noción de ataques entre conjuntos de suposiciones.

Formalmente un conjunto de suposiciones A es:

- *Admisible* si y sólo si no se ataca a sí mismo y ataca a todos los conjuntos de suposiciones que lo atacan;
- *Preferido* si y sólo si es maximal.
- *Escépticamente preferido* si y sólo si es la intersección de todos los conjuntos de suposiciones preferidos;
- *Completo* si y sólo si es admisible y contiene todas las suposiciones que defiende, donde A defiende a a si y sólo si A ataca a todos los conjuntos de suposiciones que atacan a a ;
- *Grounded* si es minimal;
- *Ideal* si y sólo si es maximal, admisible y contenido en todos los conjuntos de suposiciones preferidas;
- *Estable* si y sólo si no se ataca a sí mismo y ataca a todas las suposiciones que no contiene.

En ABA hay diversas semánticas disponibles (crédulas, escépticas, grounded), y la decisión de cuál usar depende mucho del problema que se quiera resolver. Algunas semánticas son más crédulas en el sentido que sancionan una gran cantidad de argumentos como ganadores, mientras que otras son más escépticas. En un entorno de toma de decisiones una semántica crédula puede ser conveniente cuando sea necesario tomar una decisión en todas las circunstancias, pero cuando la decisión es demasiado crítica o controvertida es mejor ser escéptico.

Ejemplo

Sea $\langle L, R, A \rangle$ un ABA framework, donde:

$L = \{a, b, c, p, q, r, s, t\}$, $R = \{p \leftarrow q, a; q \leftarrow r; r \leftarrow b, c\}$, $A = \{a, b, c\}$, y se cumple que $\bar{a} = r$; $\bar{b} = s$; $\bar{c} = t$.

Luego, a partir de estos elementos se obtiene:

$$\{\} \vdash q, \{a\} \vdash p, \{b, c\} \vdash r, \{a\} \vdash a, \{b\} \vdash b, \{c\} \vdash c$$

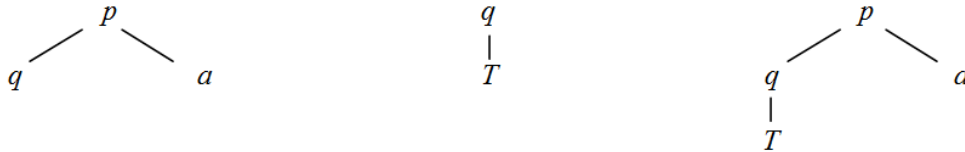


Figura 2.5: Deducciones para p (izquierda), q (centro) y p (derecha) para el ejemplo presentado en la sección “ABA frameworks”. El símbolo T representa el valor de verdad verdadero.

Los argumentos pueden representarse en forma de árbol, tal como se muestra en la figura 2.5. En este ejemplo en particular, se cumplen las siguientes relaciones de ataques:

$$\{b, c\} \vdash r \text{ ataca } \{a\} \vdash p \text{ al igual que a } \{a\} \vdash a.$$

Diversas semánticas están disponibles en ABA (e.g. grounded skeptical semantics, admissible semantics, etc.). En esta Tesis se utiliza la semántica *grounded skeptical*, ya que es la que sanciona solamente un conjunto de argumentos como “ganadores”. En [Ton14] se proporciona un análisis más profundo del framework en cuestión.

2.3. Resumen del capítulo

En este capítulo se definieron los fundamentos teóricos y matemáticos que serán utilizados en capítulos subsiguientes. Entre ellos, se presentaron conceptos vinculados a redes P2P y el framework ABA. El concepto de red P2P es fundamental en el marco de esta Tesis, por lo que se brindó una definición concisa junto con una clasificación. Las métricas utilizadas para analizar la topología de la red lógica, en especial el coeficiente de clustering, son utilizadas en el Capítulo 5 como mecanismo de evaluación. Por este motivo, se presentaron definiciones apropiadas junto con sus correspondientes referencias bibliográficas para una profundización del tema. Por último se presentaron nociones básicas de argumentación bajo los fundamentos del framework Assumption-Based Argumentation (ABA), utilizado en la parte final de esta Tesis como mecanismo deductivo para optimizar las búsquedas en los sistemas P2P.

Capítulo 3

Algoritmos de Ruteo: una visión en perspectiva

En este capítulo, se presenta una revisión de literatura en donde se detallan y comparan soluciones al problema de recuperación de información en sistemas P2P. Mientras que existen algoritmos que han facilitado la implementación de arquitecturas robustas, aún tienen que superar distintas limitaciones impuestas por los mecanismos de búsqueda actuales. Hoy en día, el problema de la recuperación de información es mucho más complejo que el problema tradicional de la búsqueda de recursos basada en identificadores de objetos o nombres. A través de los años, la comunidad de recuperación de la información ha desarrollado una gran cantidad de técnicas para la búsqueda centralizada. Más allá de eso, estos métodos no pueden ser directamente aplicados a la recuperación de información en redes P2P ya que la búsqueda en este nuevo contexto deja de ser centralizada. Dada la explosión de información que se dio en las últimas décadas, desarrollar estas nuevas habilidades es un paso importante para desarrollar aplicaciones más eficientes y brindar la posibilidad de desarrollar otras nuevas para darle más funcionalidades a las redes P2P y dejar de verlas sólo como un simple medio de almacenamiento. Una revisión de algunos de los primeros métodos desarrollados para solucionar estos problemas se presenta en [ZYKG04], mientras que una revisión más recientes es presentada en [THT12].

Este capítulo se organiza de la siguiente forma: En la sección 3.1 se presentan los algoritmos de búsqueda en redes P2P más relevantes, dando énfasis a aquéllos que utilizan aspectos semánticos. En la sección 3.2 se brinda una detallada comparación y se propone una posible clasificación para los algoritmos descritos en la sección 3.1. Se concluye el

capítulo con la sección 3.3 donde se brinda un breve resumen de los aspectos más relevantes del capítulo.

3.1. Algoritmos de ruteo

En esta sección se presenta una revisión de los algoritmos de ruteo más importantes en redes P2P *estructuradas*, *semi-estructuradas* y *no estructuradas*, siendo estas últimas el foco central de atención. En cada caso, se presenta una descripción del algoritmo junto a la bibliografía correspondiente.

3.1.1. Algoritmos de ruteo en redes P2P estructuradas

En un sistema estructurado, las relaciones entre los pares está predefinida, lo que facilita la localización del contenido. La búsqueda en este tipo de sistemas se ve determinada por la topología física de la red. La gran mayoría de los sistemas estructurados utilizan tablas hash distribuidas (DHT) para la localización del contenido. Los sistemas que utilizan DHT pueden tener estructuras planas o jerárquicas, siendo uno de sus principales beneficios la posibilidad de explotar la estructura física de la red para garantizar algoritmos de búsqueda de un determinado tiempo de ejecución. No es natural implementar un algoritmo con DHT en una red no estructurada debido a su falta de estructura, aunque más allá de esta observación, existen autores que han explotado su uso en sistemas no estructurados y semi-estructurados [Her05].

Las estructuras planas incluyen anillos, mallas, hipercubos y algunos tipos especiales de grafos como el “grafo de Bruijn” [Bru46]. A modo de ejemplo, *Chord* [SMK⁺01] usa una estructura de anillo conformada por los identificadores (IDs) de los nodos. Cada nodo mantiene una tabla que contiene las direcciones IP de aquéllos que se encuentran a la mitad de la distancia respecto su propio ID. Una clave es mapeada con un nodo cuyo ID es el mayor número que no excede el valor de esa clave. Durante el proceso de búsqueda, un nodo A reenvía la consulta con la clave k hacia $\text{sucesor}(k)$, que es otro nodo que se encuentra en la tabla de A con el mayor ID que no sobrepasa el valor de k . De esta manera la consulta para la clave k se reenvía a través de la lista de sucesores hasta que se alcanza el nodo responsable de la clave k . Se genera así lo que suele denominarse “finger

table” que acelera notoriamente la velocidad de la búsqueda, garantizando un tiempo de ejecución de $O(\log N)$.

Pastry [RD01] utiliza una estructura basada en árboles, que puede considerarse una generalización de un hipercubo. Cada nodo A mantiene un conjunto L de hojas. Para cada nodo A , el conjunto L consiste de aquellos $L/2$ nodos cuyos ID son los más cercanos y menores que el ID de A , junto con el conjunto de $L/2$ nodos cuyos IDs son cercanos y mayores que el ID de A . Este conjunto de hojas garantiza la correctitud del proceso de ruteo. Para disminuir la latencia, cada nodo *Pastry* mantiene una tabla de ruteo con punteros hacia otros nodos que se encuentran en el mismo espacio de identificadores (ID space). La búsqueda en *Pastry* se realiza de la siguiente manera: dada una consulta para una clave k , un nodo A reenvía la consulta hacia otro nodo cuyo ID es numéricamente el más cercano a k entre todos los nodos que A conoce. Este nodo en un principio trata de encontrar otro nodo en su lista de hojas. Si ese nodo no existe, el nodo A busca en su tabla de ruteo algún nodo cuyo ID comparta el mayor prefijo con k respecto al que comparte con A . Si este nodo tampoco existe, el nodo A reenvía la consulta a aquellos nodos cuyos IDs tienen el mismo prefijo que A pero son numéricamente más cercanos a k que A . El concepto de proximidad en la red puede ser considerado usando información heurística durante el reenvío de las consultas en *Pastry*. Cada nodo *Pastry* asegura de esta manera un tiempo de ejecución de $O(\log N)$. El enfoque presentado en [ZHS⁺04] llamado *Tapestry* es similar a *Pastry*. Estos dos sistemas difieren en el algoritmo de ruteo subyacente y en el enfoque adoptado para explotar la localidad. *Tapestry* también asegura un tiempo de ejecución de $O(\log N)$.

En *CAN* [RFH⁺01] se utiliza un espacio d -dimensional para implementar las tablas hash distribuidas. El espacio se divide en diferentes zonas (siendo cada una de ellas un hiper-rectángulo) y cada una de estas zonas es controlada por un determinado nodo. Una clave k es mapeada con un punto p en el espacio d -dimensional. Los nodos que son cubiertos por el punto p tienen en su tabla hash una entrada para la clave k . La tabla de ruteo de cada nodo consiste de todos sus vecinos en el espacio d -dimensional. Un nodo A se considera vecino de otro nodo B si la zona de B comparte un hiperplano $(d - 1)$ -dimensional con la zona de A . Dada una consulta con clave k , un determinado nodo reenvía la consulta hacia otro nodo que tiene en su tabla de ruteo cuya zona es la más cercana a la zona del nodo responsable de la clave k .

KaZaA [KaZ] es un sistema P2P cuya finalidad es compartir archivos, principalmente

archivos de música en formato mp3 a través de Internet. Este sistema utiliza una jerarquía de dos niveles, seleccionando como *super-nodos* a aquéllos con mayor velocidad de conexión a Internet y mayor capacidad de CPU. Cada super-nodo indexa los archivos que se manejan en su grupo. Dado que estos super-nodos son más veloces y con mejores conexiones que el resto, se mejora el rendimiento general del sistema respecto a aquéllos que utilizan un enfoque de índices locales. La contrapartida es el costo que genera el intercambio de información entre los nodos hojas y los super-nodos [YDRC06, LC06].

En [HCHL05] se presenta un algoritmo de ruteo estructurado en dos niveles: *SkipNet* y *Small World*. El primer nivel rutea las consultas basándose en un ID numérico y el segundo usando una topología de mundo pequeño (descrita en el capítulo 2).

pSearch [TXM03] es un eficiente sistema de recuperación de información en redes P2P que soporta búsquedas basadas en la semántica del texto. El sistema *pSearch* evita el problema de la escalabilidad que presentan algunos sistemas que utilizan índices centralizados. También evita el problema del no determinismo que está comúnmente presente en los sistemas basados en heurísticas. En *pSearch*, los documentos en la red están organizados respecto a su representación vectorial (basado en algoritmos de ranqueo) y el espacio de búsqueda para una consulta es organizado de acuerdo a los documentos más relevantes. La idea fundamental en *pSearch* es almacenar la información de los documentos en las DHT de acuerdo a sus representaciones vectoriales, de tal manera que el espacio de búsqueda para una consulta determinada se organice en torno a los documentos más relacionados.

Un novedoso enfoque que utiliza un framework híbrido es presentado en [YGS15]. La adaptabilidad de este framework está dada por su capacidad reactiva y pro-activa para descubrir contenido. El framework utiliza la identificabilidad del contenido y prefijos para clasificar los datos sensibles al tiempo con el propósito de aplicar la mejor estrategia para cada categoría. Para la estrategia pro-activa los autores proponen un algoritmo de ruteo jerárquico denominado “Bloom-Filter” (ver [BM04] para tener una revisión detallada de distintos aspectos de bloom-filters). En este caso el algoritmo es estructurado y tiene una estructura geográfica auto-organizable de modo tal que permite que este enfoque sea escalable a grandes redes tal como la “metropolitan Vehicular Ad-Hoc Networks” (VANETs).

El enfoque presentado en [LKG13] propone organizar un enorme y caótico espacio de búsqueda en múltiples sub-espacios basados en la semántica del contenido. Los participantes en cada sub-espacio comparten intereses semánticamente similares, formando de este

modo organizaciones virtuales basadas en intereses semánticos (Virtual Organizations-VOs). De este modo las búsquedas pueden realizarse en cada VO y las consultas pueden ser rápidamente propagadas hacia miembros adecuados dentro de las VOs. Para organizar los distintos intereses y facilitar la construcción de las VOs, los autores proponen un modelo ontológico abstracto y genérico que guía a los usuarios para determinar las propiedades ontológicas y elegir la VO adecuada. Para implementar el directorio de ontologías en forma descentralizada, se usaron DHT que indexan una taxonomía jerárquica. Aunque el modelo basado en ontologías facilita la formación de las VOs, las búsquedas siguen siendo un desafío dada la naturaleza heterogénea, distribuida y dinámica que presenta el espacio de búsqueda. Por este motivo se presenta una infraestructura llamada *OntoSum*, para descubrir recursos dentro de las VOs en forma eficiente.

3.1.2. Algoritmos de ruteo en redes P2P semi-estructuradas o débilmente estructuradas

En este tipo de sistemas, la estructura de la red no está estrictamente especificada. La estructura emergente se crea en forma probabilística o es determinada por alguna estructura topológica subyacente. La búsqueda en este tipo de redes depende de la estructura subyacente y de cómo la información es almacenada [SMwW⁺03].

FreeNet [CSWH01, FRE11] es un sistema descentralizado débilmente estructurado creado con el objetivo de mantener el anonimato de las fuentes de los datos. Este sistema está implementado utilizando una interfaz DHT. Cada nodo mantiene un directorio local de datos y una tabla de ruteo dinámica. La tabla de ruteo de un nodo contiene las direcciones de otros nodos y las posibles claves que esos otros nodos mantienen en sus propias tablas. Dada una consulta por un determinado archivo con clave k , el nodo demandante A busca en su directorio local. Si el archivo es encontrado localmente, se da por resuelta la consulta. En caso contrario, A reenvía la consulta al nodo B , tal que B se encuentra en la tabla de ruteo de A y tiene la clave más cercana a k . Cuando B recibe la consulta se realiza un procedimiento similar; si el archivo no se encuentra en el repositorio local de B entonces este nodo vuelve a reenviar la consulta a aquel nodo que en su tabla tenga la clave más cercana a k . Esta sucesión de re-envíos continúa hasta que el proceso de búsqueda termina. Durante el ruteo, puede suceder que un nodo no re-envíe la consulta al nodo con clave más cercana porque el nodo en cuestión se puede encontrar

caído o se puede haber detectado un ciclo. En estos casos el nodo trata de conectarse con el segundo nodo que tenga la clave más cercana. Si no hay ningún nodo que cumpla con estas condiciones, el nodo reporta una falla al nodo que originó la consulta (nodo demandante). Un parámetro *TTL* (time to live) es especificado para establecer un límite en el número de re-envíos durante el ruteo. Cuando el archivo es encontrado, es devuelto al nodo demandante recorriendo el camino inverso al de la consulta.

PHIRST [RGKK08] es un sistema que facilita la búsqueda efectiva de texto en bases de datos P2P. Este enfoque maneja en forma efectiva los aspectos más relevantes de sistemas estructurados y no estructurados. En forma similar a los enfoques estructurados, los nodos poseen claves para los documentos que almacenan (estas claves en realidad son términos). Sin embargo, los términos frecuentes se identifican de forma rápida y no se almacenan de manera exhaustiva, lo que resulta en una reducción significativa en los requerimientos de almacenamiento del sistema. Durante el proceso de búsqueda se utiliza un enfoque no estructurado para poder localizar aquellos términos menos frecuentes. Esta combinación de ambos enfoques permite lograr un balance de costos que lleva a una reducción significativa en la sobrecarga en el sistema, la cual es generada por el hecho de encontrar respuesta a una consulta dada.

Existen otros tipos de redes P2P semi-estructuradas donde la red se encuentra dividida en sub-redes, resultando en una topología basada en los intereses de los nodos miembros de la red. En [MDPY13] se presenta un sistema donde los nodos son agrupados de acuerdo a sus intereses [BGSWW14], obteniendo de este modo sub-redes P2P que forman un dominio multi-capas basado en intereses tal como se muestra en la figura 3.1. En este sistema existen tres tipos de nodos: los *nodos activos*, los *super nodos* y los *nodos normales*. Cada nodo activo actúa como un router, brindando información que facilita el ruteo de las consultas a nivel de grupo o de sub-red. Un super nodo es un nodo representativo de un grupo y es el encargado de verificar que los nodos del grupo compartan intereses entre sí. Por último, un nodo normal es el principal responsable de la prestación de diversos tipos de recursos compartidos.

Un enfoque similar se presenta en [UAO13], donde sus autores proponen un sistema en el que se crean grupos en múltiples capas lógicas. En este sistema, los usuarios pueden cambiar de red o de capa (o nivel) en la red para buscar contenido en función de su popularidad. Una de las redes subyacentes es una red basada en clusters construidos a partir del contenido de los pares [UO12, OU11].

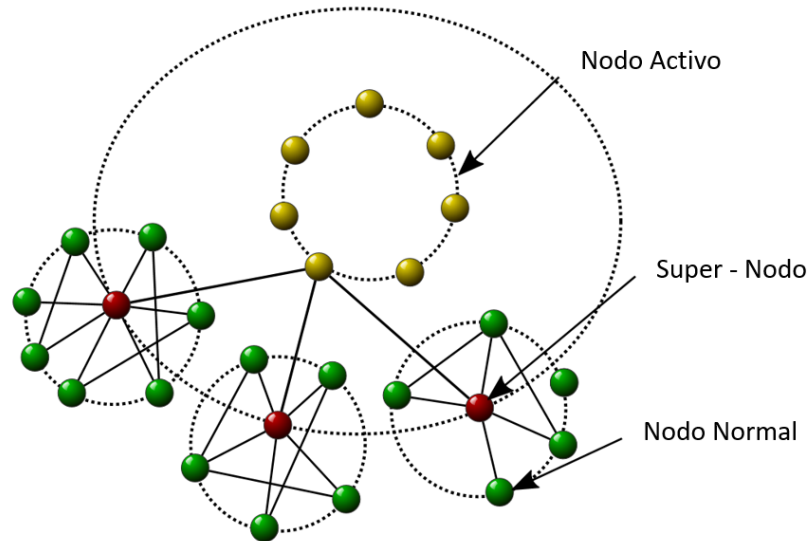


Figura 3.1: Estructura topológica de una red P2P con múltiples capas. Gráfico adaptado de [MDPY13].

Un método que modela los intereses de los usuarios en un sistema P2P cuyo objetivo es compartir documentos se presenta en [QYL14]. En este enfoque se crea una red subyacente basada en el algoritmo de clustering k -medoids, que se combina con el historial de consultas de los usuarios para mejorar el modelo de búsqueda.

3.1.3. Algoritmos de ruteo en redes P2P no estructuradas

En una red P2P no estructurada, no existen reglas que definan en forma estricta dónde se encuentra almacenada la información y qué nodos son vecinos entre sí. El algoritmo *Breadth First Search* (BFS) o *Flooding* es el algoritmo más básico utilizado para realizar búsquedas en este tipo de redes. En este algoritmo, las consultas son propagadas de un nodo hacia todos sus vecinos inmediatos, luego a los vecinos de esos nodos y así siguiendo hasta alcanzar el valor establecido por el parámetro *TTL*. Este método de ruteo es implementado en algunos sistemas como *Napster* y *Gnutella* [Nap, Rip01]. El método *Flooding* tiene por objetivo encontrar la mayor cantidad posible de resultados, aunque no es escalable y genera una gran sobrecarga en el sistema comparado con otros enfoques [Rip01].

Distintas alternativas han sido propuestas para mejorar el algoritmo básico *Flooding* en redes P2P no estructuradas. En *Iterative deepening* [YGM02], también llamado “expanding the ring”, el nodo demandante realiza en forma periódica una secuencia de búsquedas BFS incrementando el límite de la profundidad $D_1 < D_2 < \dots < D_n$. La consulta se considera resuelta cuando se encuentra una respuesta o cuando se alcanza el límite de profundidad n . En el último de los casos, la consulta no es resuelta y el nodo demandante nunca encontrará una respuesta. Todos los nodos utilizan la misma secuencia de límites dados por una lista de profundidades llamada *policy* P y el mismo período de tiempo entre dos búsquedas BFS consecutivas. Este algoritmo es apropiado para aplicaciones donde el número inicial de hits es importante, pero no reduce el número de mensajes duplicados y el tiempo asociado al procesamiento de la consulta es relativamente alto comparado con otros enfoques.

En el algoritmo *Depth-First Search* (DFS), en lugar de reenviar una consulta a los vecinos, cada nodo selecciona un único vecino candidato para hacer el reenvío. En ese caso, se utiliza el valor máximo para el parámetro *TTL* que determina la profundidad de la búsqueda. Si el nodo demandante no recibe una respuesta en un determinado período de tiempo, selecciona a otro vecino como candidato para volver a enviar la consulta. Este proceso se repite hasta que se encuentra una respuesta o todos los vecinos son seleccionados. El criterio utilizado para seleccionar un vecino puede influir fuertemente en el desempeño del proceso de búsqueda. *FreeNet* [CSWH01, FRE11] es un ejemplo de un sistema P2P que utiliza un algoritmo de búsqueda BFS.

En el algoritmo *Random-Walk* [GMS04] standard, el nodo demandante reenvía su consulta hacia un vecino seleccionado en forma aleatoria. Este vecino vuelve a repetir el proceso descrito que continúa hasta que se encuentra una respuesta. En el algoritmo *Random-Walk* standard se utiliza sólo un caminante o “walker”, lo que ayuda a disminuir la cantidad de mensajes que circulan por el sistema pero provoca que el tiempo de búsqueda sea mayor. En *k-walker Random Walk* [LCC⁺02], se utilizan k copias del mensaje que son reenviadas por el nodo demandante hacia k nodos seleccionados en forma aleatoria. Cada uno de estos mensajes sigue su recorrido en la misma manera que el enfoque original. Cada uno de estos caminantes se comunica periódicamente con el nodo demandante para decidir cuándo se debe terminar el proceso. El objetivo de este algoritmo es mejorar el tiempo de la búsqueda.

El algoritmo *Two-Level k-Walker-Random-Walk* [JW04] utiliza el mismo principio: el

nodo demandante envía k_1 caminantes con un *TTL* establecido en l_1 , cuando este *TTL* alcanza el valor cero, cada caminante envía otros k_2 caminantes con un *TTL* establecido en l_2 . Todos los nodos involucrados en el camino de los caminantes procesan la consulta. Este algoritmo genera una menor cantidad de mensajes duplicados pero el tiempo de búsqueda es mayor respecto al algoritmo *k-walker random walk*. Los enfoques “Random-walk” son populares en las aplicaciones P2P. Por ejemplo, en [LYHC14] se presenta un estudio de las limitaciones de los algoritmos estilo *Random-Walk*, presentando además un algoritmo codicioso que asegura un rendimiento óptimo.

En el algoritmo *Modified Random BFS* [KGZY02b] el nodo demandante reenvía la consulta a un subconjunto de sus vecinos que son seleccionados en forma aleatoria. Cuando ellos reciben la consulta, cada uno la vuelve a reenviar de la misma manera (excluyendo al nodo que la reenvió en el paso anterior). Este proceso termina cuando alguna condición parametrizable es satisfecha. Si bien el mecanismo del algoritmo es similar a los algoritmos que siguen el estilo utilizado en *Random-Walk*, el uso de la condición parametrizable permite controlar la cantidad de nodos que se visitan y mejorar de este modo la cantidad de consultas satisfechas.

Directed BFS [YGM02] es un algoritmo en el que el nodo demandante envía los mensajes de consulta a aquellos vecinos que probablemente le retornen los resultados de mejor calidad en menor tiempo. Luego estos vecinos vuelven a reenviar el mensaje utilizando un algoritmo BFS. Para elegir a los “mejores” vecinos, un nodo mantiene información respecto a estadísticas simples sobre sus vecinos (como el número de consultas satisfechas, la latencia, etc.). Basándose en estas estadísticas, los mejores nodos pueden ser inteligentemente seleccionados utilizando información heurística como el mayor número de resultados retornados anteriormente por un determinado vecino. Este algoritmo procesa la información sobre la historia pasada guardando datos para luego aplicar determinadas heurísticas para obtener el conjunto de nodos candidatos. Todo esto hace que este enfoque sea un método orientado a la consulta. Este algoritmo logra reducir el costo de ruteo en término de la cantidad de mensajes que son reenviados. Seleccionando los vecinos más prometedores, esta técnica logra mantener la calidad de los resultados y mejorar el tiempo de respuesta. Más allá de esto, sólo el nodo demandante selecciona el subconjunto de nodos candidatos, el resto de los nodos realiza el reenvío utilizando el algoritmo clásico BFS; de esta manera, la cantidad de mensajes duplicados no se reduce de manera significativa.

Intelligent search [KGZY02b] es un algoritmo similar a *Directed BFS*. Cuando el nodo

demandante inicia una consulta, evalúa la historia pasada de sus vecinos para propagarla sólo a un subconjunto de los mismos. Este subconjunto contiene a aquellos nodos que han respondido consultas similares en el pasado. Cuando un nodo recibe un mensaje de consulta, trata de resolverla con su contenido local. Si se encuentra la respuesta en su contenido es devuelta al nodo demandante y el algoritmo finaliza. En caso contrario, el vecino reenvía la consulta al subconjunto de sus propios vecinos que han respondido consultas similares en el pasado. La finalización de este proceso está dada por un parámetro *TTL* previamente establecido. El modelo de similitud por coseno [Hua08] es utilizado para computar la similitud entre las consultas. Para determinar qué vecinos respondieron las consultas similares, cada nodo almacena un perfil de sus vecinos a partir del cual es posible estudiar su historia pasada, lo que hace que este método sea orientado a la consulta.

En el algoritmo *Local-Index-Based Search* [YGM02] cada nodo mantiene índices sobre los datos de todos los nodos que se encuentran a una distancia de k -saltos. De esta forma, cada nodo puede obtener respuestas sobre cualquier dato que se encuentra en su índice sin necesidad de recurrir a otros nodos. Todos los nodos utilizan una misma política P sobre la lista de profundidades. Esta profundidad indica hasta cuándo una consulta debe ser procesada. Los nodos cuyas profundidades se encuentran definidas en P verifican sus índices locales y retornan el resultado buscado en caso de encontrarlo. Estos nodos también reenvían el mensaje de consulta a todos sus vecinos siempre y cuando sus profundidades no alcancen el límite máximo. El resto de los nodos cuyas profundidades no se encuentran listadas en P , simplemente reenvían la consulta a sus vecinos sin verificar si la respuesta se encuentra en su índice. El índice local se actualiza cada vez que un nodo se une a la red, cuando la abandona o cuando se modifica la información. Un nodo Y ingresa a la red enviando un mensaje de ingreso con un *TTL* de r saltos. Este mensaje de ingreso contiene meta-datos (índices) respecto a la colección de datos de Y . Todos los nodos en una distancia de r -saltos de Y reciben este mensaje. Si un nodo X recibe un mensaje de ingreso de Y , este responde con otro mensaje de ingreso con meta-datos de su propia colección de datos para mantener el índice actualizado. Cuando un nodo Z abandona la red o falla, el resto de los nodos debe detectar este evento dentro de un determinado tiempo. Si estos nodos indexan la colección de datos de Z , deben eliminar de sus índices las entradas correspondientes. Cuando la colección de datos de Z se modifica, Z envía un mensaje de actualización con un *TTL* de r -saltos a todos sus vecinos. Este enfoque es similar a *Iterative Deepening*. Ambos enfoques reenvían las consultas basándose en una lista de profundidades, aunque en *Iterative Deepening*, todos los nodos dentro de una

máxima profundidad procesan la consulta. Además en *Local-Index-Based Search*, cada nodo indexa la información de otros, mientras que en *Iterative Deepening* todos los nodos mantienen índices locales pero individuales.

Routing index based search (RI) [CGM02] es un algoritmo con enfoque parecido a *Directed BFS* e *Intelligent Search* en el sentido de que todos ellos utilizan información respecto a los vecinos para guiar las búsquedas. En *Directed BFS* sólo el nodo demandante aplica el criterio de búsqueda inteligente. El resto de los nodos realiza el ruteo de la consulta tal como en BFS. Por otro lado tanto *Intelligent Search* como *Routing Indices* realizan este proceso de selección de vecinos potencialmente útiles en todos los nodos involucrados en el proceso de búsqueda. La diferencia entre estos dos algoritmos radica en la información que guardan respecto a los vecinos. *Intelligent Search* utiliza información respecto a las consultas que fueron respondidas por los vecinos mientras que *Routing Indices* (RI) almacena información respecto a los tópicos y cantidad de documentos que se encuentran en un determinado vecino. En general, un buen vecino es aquel mediante el cual se pueden encontrar más respuestas en un menor tiempo. En este tipo de sistemas es importante mantener controlado el volumen de los índices; es por ellos que en lugar de utilizar el sistema tradicional de indexar destinos los autores utilizan RIs, en donde se almacena una ruta de donde se puede encontrar un documento en lugar de almacenar su ubicación actual.

Para ilustrar el algoritmo RI, se utiliza el ejemplo presentado en [CGM02]. En la figura 3.2 se muestran cuatro nodos *A*, *B*, *C* y *D*, conectados por líneas sólidas. El documento con contenido *x* se encuentra en el nodo *C* pero el RI (índice de ruteo) del nodo *A* está referenciando a *B* en lugar de a *C* (línea punteada). Usando rutas en lugar de destinos, la dimensión de los índices es proporcional al número de vecinos en lugar de serlo a la cantidad de documentos. El tamaño de los RIs se reduce usando índices aproximados, por ejemplo, permitiendo a los índices dar una sugerencia o pista (en lugar de la respuesta definitiva) respecto a dónde se encuentra un documento. Por ejemplo, en la misma figura, una entrada en el RI del nodo *A* debería cubrir los documentos con contenido *x*, *y* o *z*. Una consulta por un documento con contenido *x* dará lugar a una pista correcta, pero una consulta por contenido *y* o *z* no la dará. Éste es un método orientado al contenido, dado que el conocimiento de los nodos se establece a partir de nuevas conexiones (no utilizando experiencia pasada).

Algunos métodos de recuperación de información en redes P2P son adaptaciones de

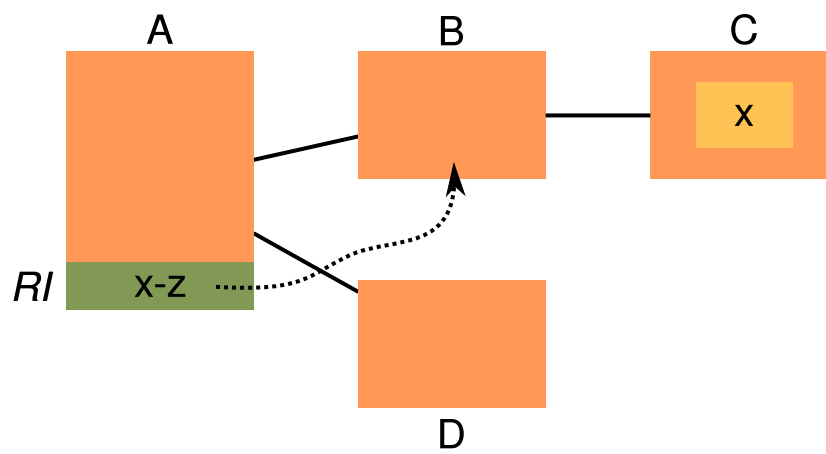


Figura 3.2: Esquema del algoritmo *Routing Indices*. Gráfico adaptado de [CGM02].

problemas de clasificación a ruteo de consultas. En un problema de clasificación, el clasificador trata de clasificar un objeto usando algunas características. El *Semantic Overlay Model* [JNCY06] tiene por objetivo localizar pares o nodos apropiados para responder a una determinada consulta. En lugar de reenviar las consultas hacia todos los vecinos, este enfoque sólo lo hace hacia nodos semánticamente similares. En este algoritmo se crean “vectores semánticos” para clasificar los nodos en categorías que representan la similitud semántica. Este es un método orientado al contenido ya que usa meta-información para clasificar a los pares por intereses. Una consulta puede ser ruteada a pares relacionados, incrementando el índice de cobertura mientras que al mismo tiempo se reduce la cantidad de saltos y de mensajes.

El método *Route Learning* [CKU09] usa palabras claves extraídas de las consultas para determinar cómo rutear las consultas. Difiere respecto a otros enfoques similares en que se utiliza meta-información para clasificar las consultas y decidir cómo rutearlas. En este método, un nodo trata de estimar los vecinos potenciales para responder a una consulta. Para calcular esta estimación, los nodos se basan en conocimiento que gradualmente fueron adquiriendo consulta tras consulta, que han recibido y enviado a sus vecinos. *Route Learning* reduce la cantidad de consultas respecto algoritmos estilo *Flooding* gracias a la extracción de los términos de las consultas para seleccionar los nodos potencialmente útiles, todo esto hace que este método sea orientado a la consulta.

Rutear las consultas usando experiencia pasada es la mejor manera de hacer el ruteo hacia nodos específicos, con el objetivo de mejorar el rendimiento. Pero también es importante manejar y almacenar en forma eficiente el conocimiento adquirido. Como consecuencia de esto, cada nodo tiene que reservar un espacio de almacenamiento determinado para guardar toda la meta-información. Este almacenamiento también implica el costo de mantenerla actualizada. El enfoque *Learning Peer Selection* [AY14] implementa un método orientado a la consulta en redes no estructuradas con la habilidad de descubrir las preferencias de los usuarios analizando su historial de descargas. El método propuesto es implementado en tres capas: la primera se especializa en almacenar y mantener actualizada la información respecto a la historia pasada. Las otras dos capas son responsables de manejar los perfiles de los usuarios para seleccionar aquellos más relevantes para enviar las consultas.

El concepto de compartir información es un tópico emergente que generalmente está basado en la cooperación entre pares. Este concepto representa el balance apropiado entre

compartir información y así mejorar el conocimiento global y el costo de enviar mensajes con esta información. El primer paso para compartir información es crear relaciones con nodos “amigos” como se implementa en *Self Learning Query Routing* (SLPS) [CGH05]. En SLPS, se aprenden los intereses de los pares de los resultados de búsquedas pasadas, lo que permite automáticamente crear relaciones de amistad basadas en la similitud de intereses. En este modelo las consultas se rutean a los nodos amigos según un patrón no estructurado. Si la búsqueda en los nodos amigos falla, se ejecuta un proceso de búsqueda estilo *Flooding*. Cada nodo califica a sus amigos dependiendo su desempeño (desde “alto” hasta “bajo”) de acuerdo al número de archivos compartidos y selecciona a los primeros k nodos del ranking. SLPS es un método orientado a la consulta ya que los pares establecen sus relaciones de amistad basándose en la experiencia pasada.

Una evaluación de diferentes estrategias de búsqueda en redes P2P basadas en el sistema *6S* [WAM05] es llevada a cabo en [AWMM06] con el propósito de mostrar el surgimiento de comunidades semánticas. Entre las estrategias de ruteo evaluadas se encuentran un algoritmo *aleatorio*, otro *codicioso* y uno de *aprendizaje por refuerzo*. Para rutear las consultas correctamente, en el algoritmo codicioso y en el de aprendizaje por refuerzo, cada par aprende y almacena los perfiles de los otros pares. El perfil de un par es definido por la información que el par mantiene para describir el contenido que almacena un determinado vecino. Adaptando la información de perfil, los pares tratan de incrementar la probabilidad de elegir adecuadamente los nodos candidatos para sus consultas. Las simulaciones demostraron que los pares pueden aprender mediante las interacciones para formar comunidades semánticas aun cuando la red es no estructurada.

INGA [LST05] es un método orientado al contenido inspirado en redes sociales y en comunidades semánticas [LST05]. En *INGA*, cada par juega un rol personal dentro una red social, mientras que localmente se almacena y procesa información que constituye el *conocimiento temático* de un nodo. Cada nodo mantiene un índice personal y juega cuatro roles diferentes: proveedor de contenidos, recomendador, nodo de arranque o nodo por defecto. Estos roles permiten a *INGA* determinar los mejores candidatos a los que una consulta debería ser enviada.

El algoritmo *Scalable Query Routing* (SQR) [KXZ05] fue formulado para lograr un bajo nivel de sobrecarga en la red durante el ruteo de las consultas. En este algoritmo, una tabla de ruteo es almacenada por cada nodo en donde se sugiere la ubicación de objetos en la red. Estas tablas están basadas en la experiencia pasada. Una estructura de

datos llamada *Exponentially Decaying Bloom Filter* (EDBF) codifica las tablas de ruteo probabilísticas de manera altamente comprimidas y permite el ruteo eficiente de consultas.

State-based (SBS) es un algoritmo orientado al contenido muy eficiente que rutea las consultas de acuerdo al estado de los nodos. Los resultados experimentales demostraron que este algoritmo mejora el rendimiento reduciendo el tiempo de respuesta y logrando un mejor balance de carga a lo largo de la red. Cada nodo SBS mantiene una lista con la información de estado de otros nodos. En SBS, la búsqueda eficiente se logra descubriendo aquellos nodos que comparten recursos deseables de acuerdo a un algoritmo de ruteo basado en lógica difusa.

Un novedoso método presentado en [SCG⁺16] busca moderar la sobrecarga de mensajes mediante un control de congestión. Este enfoque garantiza caminos alternativos para el ruteo y de este modo balancea la carga entre los pares de la red. En este caso el algoritmo es orientado al contenido y está basado en *Collaborative Q-Learning* ya que aprende parámetros de red, tales como la capacidad de procesamiento, número de conexiones, y el número de recursos en los nodos, junto con su estado de congestión.

Otros algoritmos de búsqueda en redes no estructuradas que son relevantes aunque no descriptos en este capítulo (aunque los incluiremos en la clasificación que se brinda más adelante) son: *q-pilot* [SE00], *SemAnt* [MGSN05], *Remindin'* [TSW04], *P2PSLN* [ZLF⁺05] y *NeuroGrid* [Jos02].

3.2. Análisis comparativo

En esta sección se presenta un análisis comparativo de los algoritmos descriptos anteriormente. En la tabla 3.1 se muestra la comparación entre los algoritmos de ruteo en redes P2P estructuradas. En este tipo de sistemas, el uso de DHT asegura un tiempo de ejecución en el orden logarítmico. El algoritmo utilizado por *SkipNet* y *Small-World* tiene una diferencia fundamental respecto a los otros sistemas tales como *Chord* y *Pastry*. En estos últimos el objetivo es implementar una interfaz DHT que permita mantener un balance de carga, mientras que en *SkipNet* la idea es permitir que los sistemas mantengan el contenido y la localidad del contenido usando una topología de mundo pequeño que permita aprovechar los caminos más cortos hacia nodos remotos.

Dada la falta de estructura en las redes totalmente descentralizadas, las búsquedas en las mismas se vuelve un proceso no escalable y dificultoso [Kle06], debido a esto los

		Características			
		DHT	Overlay		Estructura
			Plana	Jerárquica	
Algoritmo	Chord	•	•		Anillo
	Pastry	•	•		Arbol
	Tapestry	•	•		Arbol
	CAN	•	•		Toroidal
	KaZaA	•		•	2-niveles
	SkipNet y Small World			•	2-niveles
	pSearch	•		•	Toroidal
	HBFR			•	Geográfica
	OntoSum	•		•	Arbol

Tabla 3.1: Tabla comparativa de las principales características de los algoritmos de ruteo en redes P2P estructuradas.

algoritmos utilizan aspectos semánticos para poder optimizar el proceso de búsqueda. La tabla 3.2 muestra los aspectos semánticos presentes en cada uno de los algoritmos de búsqueda en redes no estructuradas que se describieron anteriormente. Los primeros cinco son algoritmos que siguen el modelo de *Flooding*, consecuentemente no poseen ningún aspecto semántico. En el resto de los algoritmos se comparte el objetivo de seleccionar en forma estratégica nodos candidatos para reducir la propagación de las consultas. Para hacer esto, algunos algoritmos (como Directed BFS) utilizan heurísticas, mientras que otros seleccionan a los candidatos basándose en la experiencia pasada (métodos orientados a la consulta) o analizando los perfiles de los nodos (métodos orientados al contenido). Finalmente existe un grupo de algoritmos que utilizan un clasificador para decidir cuáles son los mejores candidatos.

Algunos de estos algoritmos (como BFS, DFS y enfoques aleatorios) no utilizan aspecto semántico alguno, es por ello que son algoritmos menos informados para la propagación de consultas en una red P2P pero aún así son algoritmos de gran utilidad. Esto se puede evidenciar en la tabla 3.3, donde se observa que aunque algunos algoritmos implementen aspectos semánticos para realizar sus búsquedas también deben valerse de algún método básico para propagar las consultas. Esos mecanismos son ejecutados sobre el conjunto de nodos candidatos o cuando no existen nodos candidatos y la consulta debe propagarse utilizando un método alternativo. Otra característica presente en estos algoritmos es el uso del parámetro *TTL*, que se decrementa cada vez que un mensaje pasa de un nodo a otro. Cuando el valor de este parámetro llega al valor cero el mensaje es descartado.

		Aspectos Semánticos			
		Información Heurística	Orientado al contenido	Orientado a la Consulta	Clasificación
Algoritmo	Flooding				
	Iterative Deeping				
	Random Walk				
	K-walker Random Walk				
	Two-level K-walker Random Walk				
	Modified Random BFS				
	Directed BFS	•		•	
	Intelligent Search			•	
	Local Indices Based Search	•		•	
	Routing Indices Based Search		•		
	Semantic Overlay Model		•		•
	Route Learning			•	•
	Learning Peer Selection			•	
	Self Learning Query Routing			•	
	6S - Random				
	6S - Greedy		•		
	6S - Reinforcement Learning		•		
	q-pilot				•
	SemAnt	•			•
	REMINDIN'			•	
	P2PSLN		•		
	NeuroGrid		•		
	INGA		•		
	SQR	•			
	SBS		•		
	Collaborative Q-Learning	•	•		

Tabla 3.2: Aspectos semánticos en algoritmos para redes P2P no estructuradas.

3.3. Resumen del capítulo

En sus orígenes, Internet fue diseñado basado en los principios de la cooperación y de la buena ingeniería, y se manejaba como una red P2P pura. En este escenario descentralizado era indispensable contar con algoritmos para realizar búsquedas. A medida que la estructura de Internet se volvió más rígida, surgieron los algoritmos para búsquedas en redes estructuradas y semi-estructuradas, donde la colaboración entre los miembros de la red no es un aspecto importante. En los últimos tiempos, las redes P2P puras han ganado importancia nuevamente, desencadenando en nuevas investigaciones respecto a algoritmos que incorporan aspectos semánticos derivados de los perfiles e intereses de los usuarios. Estos aspectos semánticos, pueden ser convenientemente explotados para mejo-

		Características			
		BFS	DFS	Aleatorio	TTL
Algoritmo	Flooding	•			•
	Iterative Deeping	•			•
	Random Walk	•		•	•
	K-walker Random Walk	•		•	
	Two-level K-walker Random Walk	•		•	•
	Modified Random BFS	•		•	n/a
	Directed BFS	•			n/a
	Intelligent Search	•			•
	Local Indices Based Search	•			•
	Routing Indices Based Search		•		n/a
	Semantic Overlay Model	n/a	n/a	n/a	•
	Route Learning	n/a	n/a	n/a	•
	Learning Peer Selection	n/a	n/a	n/a	•
	Self Learning Query Routing	•			n/a
	6S - Random	•		•	n/a
	6S - Greedy	•			n/a
	6S - Reinforcement Learning	•			n/a
	q-pilot	n/a	n/a	n/a	•
	SemAnt	n/a	n/a	n/a	•
	REMINDIN'	n/a	n/a	n/a	n/a
	P2PSLN	n/a	n/a	n/a	•
	NeuroGrid	•			•
	INGA	n/a	n/a	n/a	n/a
	SQR	n/a	n/a	n/a	n/a
	SBS	n/a	n/a		•
	Collaborative Q-Learning	•			•

Tabla 3.3: Algoritmos básicos de ruteo en redes P2P no estructuradas.

rar el rendimiento de los algoritmos de ruteo, minimizando el tráfico en la red y mejorando el tiempo que se tarda en obtener una respuesta para una determinada consulta.

En este capítulo se presentaron varios algoritmos de búsqueda en redes P2P. Para facilitar el análisis de los mismos, se introdujeron diferentes clasificaciones. En particular, se discutieron diversas estrategias de búsqueda para sistemas estructurados, semi-estructurados y no estructurados. Finalmente se identificaron aspectos comunes en estos algoritmos y se presentó un análisis comparativo respecto a los aspectos más relevantes de los sistemas mencionados.

Capítulo 4

Algoritmos de ruteo en redes P2P

La Internet es una red con naturaleza cooperativa y descentralizada construida a partir de millones de participantes que almacenan y comparten grandes volúmenes de información entre sí. Los sistemas P2P van de la mano con esta gran red descentralizada, donde cada participante puede tanto brindar como requerir contenido. En este escenario el análisis, el desarrollo y testeo de algoritmos de búsqueda distribuida es una línea de investigación relevante. Dentro de los algoritmos para la búsqueda distribuida se encuentran aquellos que se basan en similitud respecto a temáticas para beneficiarse del surgimiento de comunidades semánticas que emergen de la interacción entre los participantes. Como resultado, los algoritmos inteligentes para la selección de vecinos prometedores (para dar respuesta a una dada consulta), conllevan la conformación de una red lógica con una topología específica que refleja patrones eficientes de comunicación. En este capítulo se presenta la contribución principal de esta tesis. En la sección 4.1 se introduce una serie de algoritmos desarrollados a partir de un mecanismo de aprendizaje basado en tópicos, cuyo objetivo es reducir el tráfico de mensajes en una red. Estos algoritmos fueron desarrollados en forma incremental, por lo tanto cada uno agrega alguna mejora respecto a los anteriores. La sección 4.2 presenta un novedoso enfoque en el que se combina el algoritmo inteligente más prometedor junto con un marco argumentativo que permite tomar decisiones acertadas respecto a cómo explorar y explotar la red física con el objetivo de optimizar las búsquedas. Básicamente lo que agrega el sistema argumentativo es una alternativa para alentar la formación de comunidades semánticas auto-contenidas, lo que trae un gran beneficio a la hora de realizar búsquedas temáticas. Para finalizar el capítulo se presenta en la sección 4.3 un resumen de la contribución principal de esta Tesis.

4.1. Algoritmos inteligentes para mejorar patrones de comunicación en redes P2P

En esta sección se presenta una serie de algoritmos desarrollados para ofrecer una solución al problema de búsqueda descentralizada distribuida en redes P2P. Exceptuando el algoritmo 1.0, todos los algoritmos comparten la misma estructura básica: cada nodo (participante de la red) posee una tabla TN (Temáticas-Nodos) en la que se almacena el conocimiento adquirido. Una entrada en esta tabla consiste de un tópico seguido de una colección de nodos que representan miembros de la red que están interesados en ese tópico. Se dice que un nodo está interesado en un tópico cuando posee un directorio con datos relacionados al tópico en cuestión. Más allá de que estos algoritmos compartan la misma estructura interna, la principal diferencia radica en dos aspectos principales: la política que adopta cada uno para actualizar su tabla TN y el mecanismo que utilizan para seleccionar vecinos potenciales a quienes propagar las consultas. Para ilustrar el comportamiento de los algoritmos propuestos, se utiliza la red hipotética que se muestra en la figura 4.1.

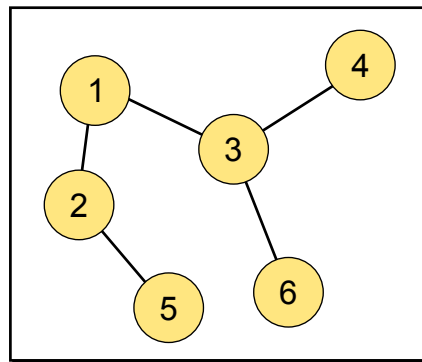


Figura 4.1: Red de ejemplo.

En el resto del capítulo, los nodos dibujados con línea de trazos son aquellos que generan una consulta, mientras que aquellos dibujados con línea punteada son los que introducen nuevo conocimiento. Respecto a las flechas, las que se muestran con línea de trazos representan el camino de un mensaje de consulta, mientras que las que se muestran con línea punteada representan el camino de un mensaje de respuesta. Por último se utiliza un tercer tipo de flecha, que se muestra mediante una línea punteada y rayada que representa el camino de un mensaje de actualización. Se utilizó una numeración básica

para nombrar a cada uno de los algoritmos, la misma permite indicar que un determinado algoritmo introduce mejoras respecto a una versión anterior. Por ejemplo, el algoritmo 2.1 es una mejora del algoritmo 2.0. Otro aspecto interesante de mencionar es que para el desarrollo de estos algoritmos se asumió que un determinado nodo sólo genera mensajes de consulta para tópicos de su interés, aunque durante las simulaciones se utilizaron mensajes esporádicos ajenos a la temática de un nodo para verificar su funcionamiento.

Algoritmo 1.0 Este algoritmo no presenta ninguna habilidad de aprendizaje, es por ello que es el único en el que los pares no almacena una tabla TN. Las consultas se rutean utilizando un patrón de fuerza bruta, como lo hace Gnutella [Rip01]. Cada vez que un nodo genera una consulta, la misma es re-enviada a todos los nodos vecinos. Este proceso se repite hasta que se alcanza una determinada cantidad de saltos previamente determinado por el parámetro *TTL*. Si uno de los nodos que recibe la consulta está en condiciones de responderla, genera un mensaje de respuesta que es enviado directamente al nodo demandante (nodo que inició la consulta). En la figura 4.2 se muestra el comportamiento de este algoritmo.

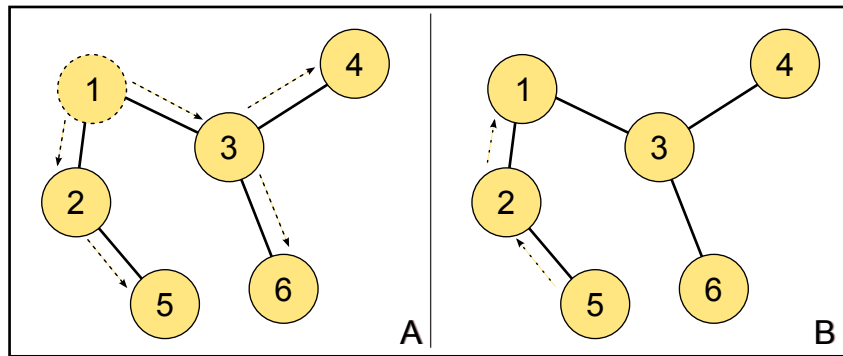


Figura 4.2: Mensajes de consulta y respuesta para el algoritmo 1.0.

Algoritmo 2.0 En este algoritmo, cuando un nodo genera un mensaje de consulta, en primera instancia verifica en su tabla TN cuáles otros nodos están interesados en la temática. Uno de estos nodos es seleccionado en forma aleatoria y luego la consulta es enviada sólo a ese nodo. En caso de que el nodo demandante no contenga en su tabla TN información respecto a nodos interesados en la temática, el mensaje de consulta se re-envía a todos los vecinos (de la misma forma que lo hace el algoritmo 1.0). La fase de aprendizaje ocurre con el uso de los mensajes de respuesta. Cuando un determinado nodo

puede responder a una consulta, genera un mensaje de respuesta que recorre el mismo camino que por el que arribó el mensaje de consulta. Cada nodo intermediario que se encuentre en ese camino, actualiza su tabla TN con la temática correspondiente y el nodo que respondió. Para ilustrar este algoritmo consideren la figura 4.3 donde se muestran los mensajes involucrados en un escenario donde el nodo 1 genera una consulta que puede ser respondida por el nodo 5, teniendo en cuenta que el nodo 1 tiene conocimiento del nodo 5 a partir de su tabla TN. Por otra parte la figura 4.4 ilustra el comportamiento del algoritmo bajo la misma circunstancia excepto que en esta situación el nodo 1 no tiene conocimiento del nodo 5.

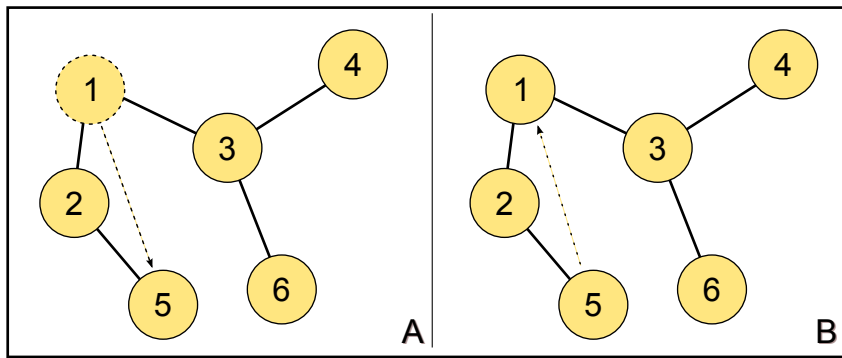


Figura 4.3: Mensajes de consulta y de respuesta para el algoritmo 2.0. El nodo 1 conoce al nodo 5.

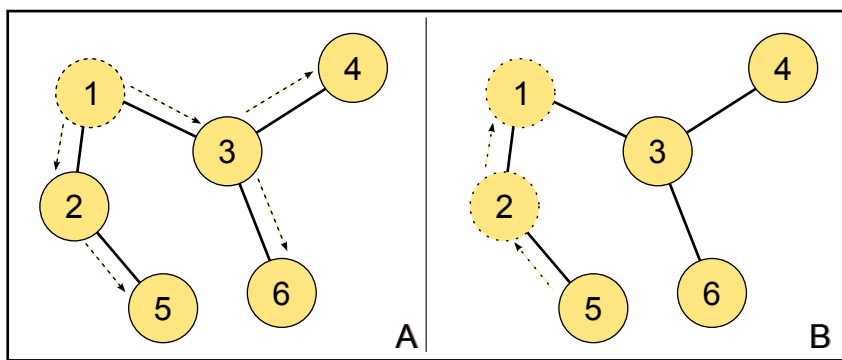


Figura 4.4: Mensajes de consulta y de respuesta para el algoritmo 2.0. El nodo 1 no conoce al nodo 5.

Algoritmo 2.1 La diferencia entre este algoritmo y el algoritmo 2.0 es que una consulta es enviada a *todos* los nodos que están interesados en la temática de acuerdo al

conocimiento almacenado en la tabla TN del nodo demandante. La fase de aprendizaje se lleva a cabo de la misma manera que en el algoritmo anterior. La figura 4.5 ilustra la situación en la que el nodo 1 genera una consulta que puede responder el nodo 5. El nodo 1 tiene conocimiento a través de su tabla TN de que los nodos 4 y 5 están interesados en la temática, por lo tanto se envía el mensaje de consulta a ambos nodos en forma simultánea (en el caso del algoritmo anterior se hubiese seleccionado sólo uno en forma aleatoria).

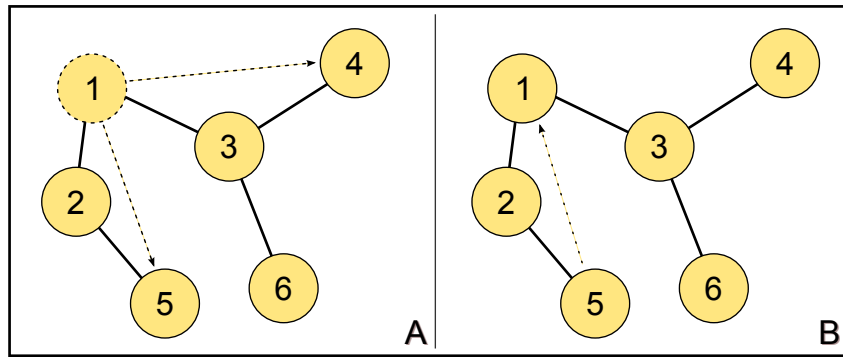


Figura 4.5: Mensajes de consulta y de respuesta para el algoritmo 2.1.

Algoritmo 3.0 El comportamiento de este algoritmo respecto a los mensajes de consulta y de respuesta es igual al del algoritmo 2.1. La diferencia radica en que este algoritmo incorpora un nuevo tipo de mensajes: los *mensajes de actualización*. Cuando un nodo incorpora nuevo conocimiento actualiza su tabla TN y luego envía un mensaje de actualización con la información recientemente incorporada —en la forma (tópico, nodo)— a todos sus vecinos. Este nuevo tipo de mensajes permite mejorar el ritmo de crecimiento de aprendizaje de la red. Mediante esta nueva versión un nodo no sólo aprende cuando se encuentra en el camino de un mensaje de respuesta, sino también lo hace cuando es vecino de un nodo que acaba de incorporar nuevo conocimiento. En la figura 4.6 se puede apreciar la situación en la que el nodo 1 no tiene conocimiento respecto al nodo 5. Los nodos que se muestran con línea punteada son aquéllos que incorporan conocimiento respecto a los intereses del nodo 5. En este ejemplo se agrega una nueva conexión a la red de ejemplo sólo a efectos de ilustrar esta situación. Se podrá notar que el nodo 2 aprende respecto a los intereses del nodo 5 por encontrarse en el camino de la respuesta y luego este conocimiento es propagado mediante un mensaje de actualización hacia el nodo 3

(vecino de 2). Finalmente el nodo 1 además de recibir respuesta a su consulta aprende que tiene temáticas en común con el nodo 5.

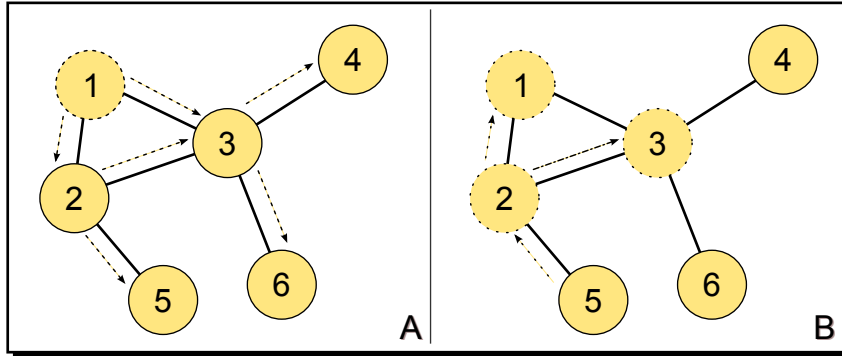


Figura 4.6: Mensajes de consulta, respuesta y actualización para el algoritmo 3.0.

Algoritmo 3.1 Esta versión difiere con el algoritmo 2.1 únicamente en el momento en que un nuevo nodo ingresa a la red. Cuando un nodo se une a la red, envía un mensaje de actualización hacia todos los vecinos directos. De esta manera, los vecinos inmediatos toman conocimiento respecto a los intereses del nuevo nodo. La figura 4.7 muestra este comportamiento mediante la situación en la que el nodo 3 ingresa como nuevo participante a la red.

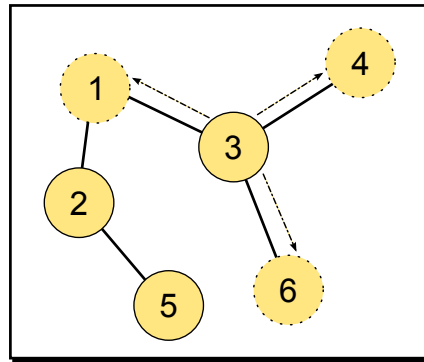


Figura 4.7: Mensajes de actualización para el algoritmo 3.1.

Algoritmo 3.2 El algoritmo 3.2 presenta una pequeña diferencia respecto al algoritmo 3.0: cuando un nodo envía un mensaje de actualización hacia sus vecinos, cada uno de ellos re-envía el mensaje original hacia sus propios vecinos. En todos los casos, se evita el re-envío del mensaje hacia el nodo que lo propagó. La figura 4.8A ilustra el primer paso: el

nodo 3 envía un mensaje de actualización a sus vecinos directos. En la figura 4.8B se puede apreciar el segundo paso: los nodos 1, 4 y 6 re-envían el mensaje a sus propios vecinos. Las flechas que se muestran dibujadas con líneas estilo raya-punto-punto representan los mensajes de actualización enviados en el primer paso mientras que las dibujadas con líneas raya-punto representan a los mensajes de actualización del segundo paso.

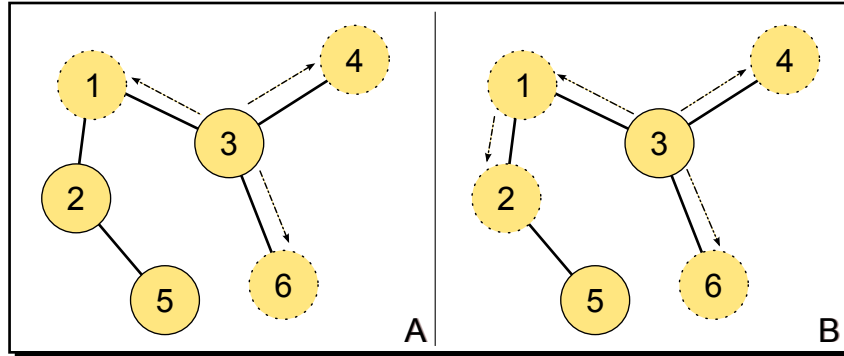


Figura 4.8: Mensajes de actualización para el algoritmo 3.2.

Algoritmo 3.3 Este algoritmo incorpora los siguientes cambios respecto a la versión 3.0: cuando un nodo recibe un mensaje de respuesta con una temática de su interés envía un mensaje de actualización a todos sus nodos vecinos y además a todos aquellos nodos que están interesados en ese tópico (según el conocimiento almacenado en su tabla TN). Este algoritmo permite que el conocimiento no sólo lo adquiera un nodo puntual sino que toda su comunidad. La figura 4.9 muestra este comportamiento: el nodo 3 está interesado en una temática en particular y recibe un mensaje de respuesta asociado a esa temática. El nodo 3 tiene conocimiento de que los nodos 2 y 5 son de su propia comunidad (tienen intereses en común), es por ello que cuando el nodo 3 recibe el mensaje de respuesta, además de adquirir nuevo conocimiento, envía mensajes de actualización con esta nueva información hacia sus vecinos y hacia los nodos de su comunidad (nodos 2 y 5).

Algoritmo 4.3.3 Este algoritmo mantiene el mismo comportamiento que el anterior además de nuevos casos en los que se utilizan mensajes de actualización. Cuando un mensaje de consulta arriba a un nodo por *broadcast* y el nodo está interesado en el tópico de la consulta pero no la puede contestar, se envía un mensaje de actualización hacia el nodo que originó la consulta con su id y sus temáticas de interés. Con este nuevo comportamiento se busca no excluir a aquellos nodos que no tienen demasiados

4.1.1. Principales características de los algoritmos

Los algoritmos presentados anteriormente comparten el mismo objetivo: optimizar las búsquedas en sistemas P2P. Cada uno de ellos presenta propiedades distintivas caracterizadas por la forma en la que realizan estas optimizaciones. Para comparar estos algoritmos se presenta la tabla 4.1 en la que se pueden apreciar las principales características de cada uno de los algoritmos.

Características	Algoritmo								
	1.0	2.0	2.1	3.0	3.1	3.2	3.3	4.3.3	5.0
Consultar a los nodos vecinos (siempre)	✓	-	-	-	-	-	-	-	-
Consultar a los nodos vecinos (sólo sin conocimiento)	-	✓	✓	✓	✓	✓	✓	✓	✓
Consultar a la comunidad semántica	-	✓ ^a	✓	✓	✓	✓	✓	✓	✓
Actualización a los nodos vecinos	-	-	-	✓	-	✓ ^b	✓	✓	-
Actualización a la comunidad semántica	-	-	-	-	-	-	✓	✓	✓
Actualización al unirse a la red	-	-	-	-	✓	-	-	-	-
Actualización al nodo demandante	-	-	-	-	-	-	-	✓	✓

^a Sólo a un miembro seleccionado aleatoriamente.

^b Dos niveles de propagación.

Tabla 4.1: Resumen comparativo de las principales características de los algoritmos.

Los algoritmos propuestos implementan las características esenciales de una gran cantidad de sistemas P2P, abstrayéndose de pequeños detalles y cuestiones de parametrización, con el objetivo de hacer foco principalmente en las características más sobresalientes. La versión 1.0 implementa un algoritmo estilo *Flooding* que es el implementado por el sistema *Gnutella* [Rip01], donde las consultas son re-enviadas hasta alcanzar un determinado *TTL* utilizando un recorrido a lo ancho del grafo que conforma la red. En el resto de los algoritmos propuestos las características más sobresalientes son el uso de tablas de ruteo y la explotación del concepto de localidad semántica. Eso se encuentra directamente relacionado con los métodos de búsqueda populares basados en índices de ruteo [CGM02]. Estos métodos utilizan información sobre los vecinos que los participantes almacenan en sus tablas de ruteo respecto a los vecinos, determinando de este modo cómo se llevará a cabo el proceso de búsqueda. En particular, el enfoque *Learning Peer Selection* [AY14]

implementa un método con la capacidad de descubrir las preferencias de los usuarios analizando su historial de descargas, dando lugar a perfiles de usuarios que son almacenados en tablas. Si la información almacenada en las tablas no es suficiente para guiar el proceso de búsqueda se utiliza el método implementado por el algoritmo 1.0.

La red lógica emergente de los algoritmos propuestos está construida a partir de las relaciones que se han creado a partir de la interacción de los nodos, lo cual es similar a lo que ocurre con el método *Self Learning Query Routing* [CGH05] que se fundamenta en establecer relaciones de amistad entre los pares. En este algoritmo en particular los intereses de los pares son aprendidos utilizando un historial de resultados de búsquedas, lo que permite construir automáticamente relaciones de amistad basándose en la cantidad de *hits* y en los intereses de los nodos en cuestión.

El concepto de “compartir conocimiento” es un tópico emergente generalmente basado en la cooperación entre pares o miembros de una red. Existe un *trade-off* fundamental entre compartir conocimiento que permita incrementar el conocimiento general de la red y el costo que involucra enviar los mensajes con esta información. Algunos de los algoritmos propuestos en esta Tesis (3.0-5.0) implementan distintas variantes de este concepto mediante el uso de mensajes de actualización. Mediante el desarrollo y testeo de los algoritmos se ha detectado el *Problema de las Comunidades Cerradas*: el hecho de compartir conocimiento lleva al surgimiento de comunidades semánticas, aunque muchas veces se evidencia que hay comunidades que tienen los mismos intereses pero que aún así se encuentran divididas. Este problema surge a raíz de que muchas veces los nodos de la red que cuentan con una menor cantidad de recursos quedan relegados. Para solucionar este problema se propone un enfoque argumentativo que es introducido en detalle en la siguiente sección.

4.2. ArgP2P: combinando comportamientos reactivos y argumentativos

Como se introdujo anteriormente uno de los problemas más complejos en redes P2P se basa en cómo realizar la búsqueda de un contenido específico. La búsqueda temática es una estrategia de resolución de este problema, brindando una forma de buscar la información según su temática. Los algoritmos que implementan la búsqueda temática se

benefician del surgimiento de comunidades semánticas, que son el resultado de la interacción entre los participantes [AWMM06]. En esta sección se presenta un novedoso modelo de búsqueda temática en sistemas P2P donde cada nodo en la red tiene la habilidad de llevar a cabo un proceso de ruteo de consultas inteligente combinando dos tipos de comportamientos: un comportamiento *reactivo* que se lleva a cabo mediante el algoritmo 4.3.3 y un comportamiento *argumentativo* que permite refinar la decisión tomada por el comportamiento reactivo.

A modo de recordar lo presentado en la sección anterior, tanto los nodos como las consultas están asociados a temáticas. De esta manera, cada nodo tiene la habilidad de analizar el tópico de un mensaje de consulta entrante y determinar si es capaz de contestar a dicha consulta o bien re-enviarla hacia otros nodos. Dada una consulta Q , el comportamiento reactivo se corresponde al modelado por el algoritmo 4.3.3 que permite tomar una decisión respecto a cómo proceder con una consulta Q que arriba a un determinado nodo. Por otra parte, el comportamiento argumentativo está basado en el conocimiento dinámico capturado por los nodos durante su ciclo de vida. De esto resulta una base de conocimiento para cada nodo que puede estar incompleta y ser potencialmente inconsistente. De esta manera, el razonamiento argumentativo (formalizado mediante argumentación basada en suposiciones [DKT09]) brinda a cada nodo la habilidad de tomar decisiones acertadas sin la intervención de otros pares. Todo esto compone el framework *ArgP2P* que extiende el modelo tradicional de toma de decisiones en sistemas P2P de modo que cada par puede actuar como un agente autónomo dotado de un sistema de inferencia que combina un comportamiento reactivo con otro puramente argumentativo. Este framework cuenta con un parámetro configurable denominado *ttarg* (por sus siglas en inglés, Time-To-Argue) que permite establecer la frecuencia en la que un determinado nodo se comporta de manera reactiva o argumentativa. Dado que la argumentación es un proceso computacionalmente costoso, el uso del parámetro *ttarg* es crucial para asegurar que el tiempo de respuesta de un nodo se mantendrá dentro de parámetros aceptables, lo que es sumamente importante en un sistema P2P.

ArgP2P ofrece una alternativa para sobrellevar el *Problema de las Comunidades Cerradas* en donde a medida que la interacción entre los pares aumenta, los mismos crean comunidades cerradas en donde sólo se comunica un limitado número de pares entre sí. De esta manera, uno o más nodos pueden quedar aislados de sus respectivas comunidades o hasta incluso pueden llegar a formar otra comunidad separada aunque tengan temáticas

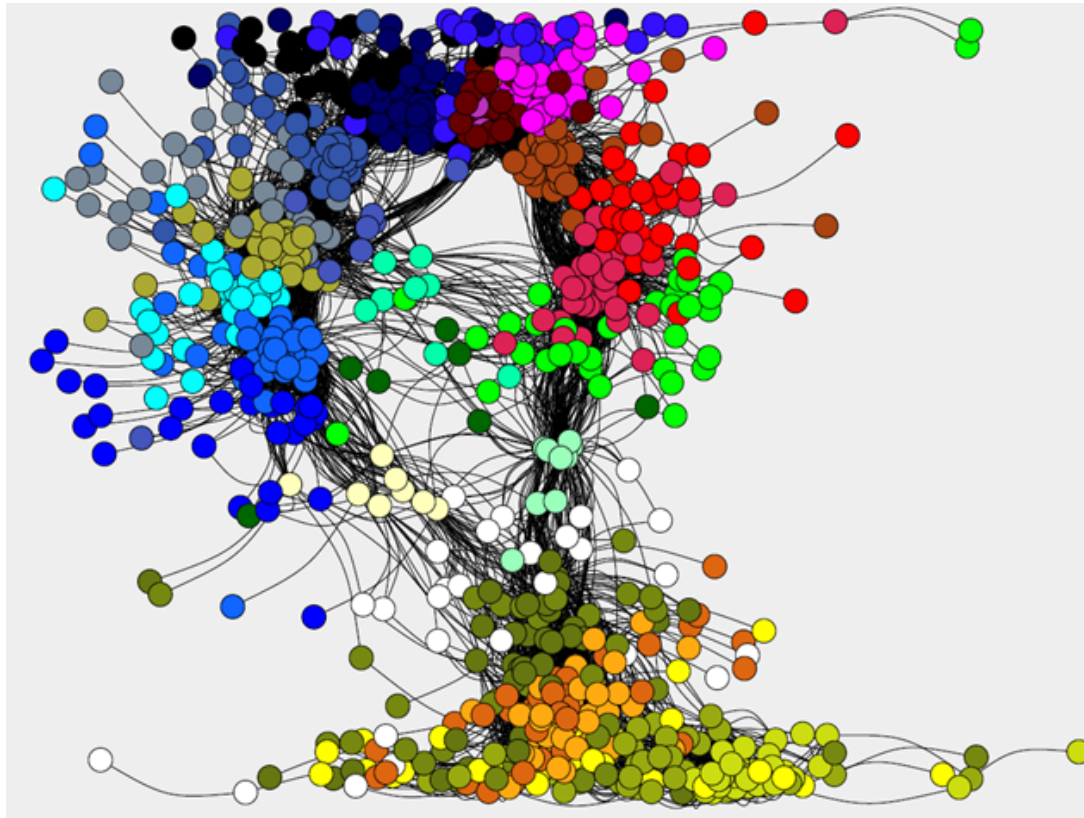


Figura 4.11: Red lógica que permite evidenciar el Problema de las Comunidades Cerradas.

en común con otra ya existente. Este problema desfavorece a aquellos pares que poseen recursos limitados o por alguna cuestión permanecen apagados gran parte del tiempo. En la figura 4.11 se puede observar una red lógica resultante de la ejecución del algoritmo 4.3.3 en la que se evidencian comunidades cerradas. En este tipo de gráficos los colores representan tópicos (colores similares representan tópicos similares) y las uniones entre los nodos representa que “un nodo conoce a otro”. En nuestro modelo un nodo puede estar interesado en más de una temática. Sin embargo, para simplificar el problema, dichas temáticas deberán estar relacionadas y en ese caso el color utilizado será la combinación de los colores correspondientes.

El objetivo de introducir un proceso argumentativo en el framework *ArgP2P* es brindar a los nodos la capacidad de decidir, de manera autónoma, si es un buen momento para explorar la red más allá de su comunidad. El propósito de llevar a cabo esta exploración es descubrir otros pares potencialmente útiles que hasta el momento eran desconocidos, ayudando de esta manera a unir comunidades o nodos, que de otra manera permanecerían aislados del resto de la red.

El resto de esta sección se organiza de la siguiente manera: en la sub-sección 4.2.1 se propone una representación formal de una red P2P, lo que brinda las bases para definir el framework *ArgP2P* que es presentado en la sub-sección 4.2.2. Para finalizar, la sub-sección 4.2.3 presenta un caso de estudio cuyo objetivo es mostrar una situación en la que la decisión tomada por el sistema argumentativo permite descubrir un nodo aislado de su comunidad.

4.2.1. Formalizando la noción de red P2P

Una red P2P es un conjunto interconectado de dispositivos que intercambian información. La representación estática de una red P2P se corresponde con la representación formal de un grafo no dirigido. Para representar las características *dinámicas* de la red se presenta una función especial para expresar el tiempo. Las características dinámicas de la red están basadas en todo lo que involucra el tráfico de mensajes en el sistema. La siguiente definición formaliza este concepto:

Definición 1 Una red P2P dinámica (o simplemente red P2P) es una terna $P = (\mathcal{N}, \mathcal{E}, S_t)$ donde \mathcal{N} es un conjunto finito de nodos, \mathcal{E} es una colección finita de arcos (cada arco es un par de elementos en \mathcal{N}) y una función S_t que captura el estado dinámico de un nodo $n \in \mathcal{N}$ en un determinado tiempo t , que se define de la siguiente manera:

$$S_t : \mathcal{N} \longrightarrow \text{Consultar} \cup \text{Reenviar} \cup \text{Responder} \cup \{\text{descartar}, \text{ocioso}\},$$

donde $\text{Consultar} =_{\text{def}} \{\text{consultar}\} \times 2^{\mathcal{N}}$, $\text{Reenviar} =_{\text{def}} \{\text{reenviar}\} \times 2^{\mathcal{N}}$ y $\text{Responder} =_{\text{def}} \{\text{responder}\} \times \mathcal{N}$.

En una red P2P un nodo posee un número finito de estados posibles: **consultar** representa el estado en el que un nodo se encuentra enviando un mensaje de consulta hacia un conjunto de nodos candidatos; **responder** representa el estado en el que un nodo se encuentra enviando un mensaje de respuesta a un determinado nodo; **descartar** indica que el nodo en cuestión descarta un mensaje; y por último cuando un nodo no se encuentra en ninguno de los estados previamente mencionados se dice que ese nodo se encuentra **ocioso**. Se puede observar que cada uno de estos estados se corresponde con un elemento de la imagen de S_t . Se asume que la red P2P en su totalidad (es decir, cada nodo de la red) se inicializa en un tiempo $t = 0$. En cualquier tiempo t , una consulta en particular

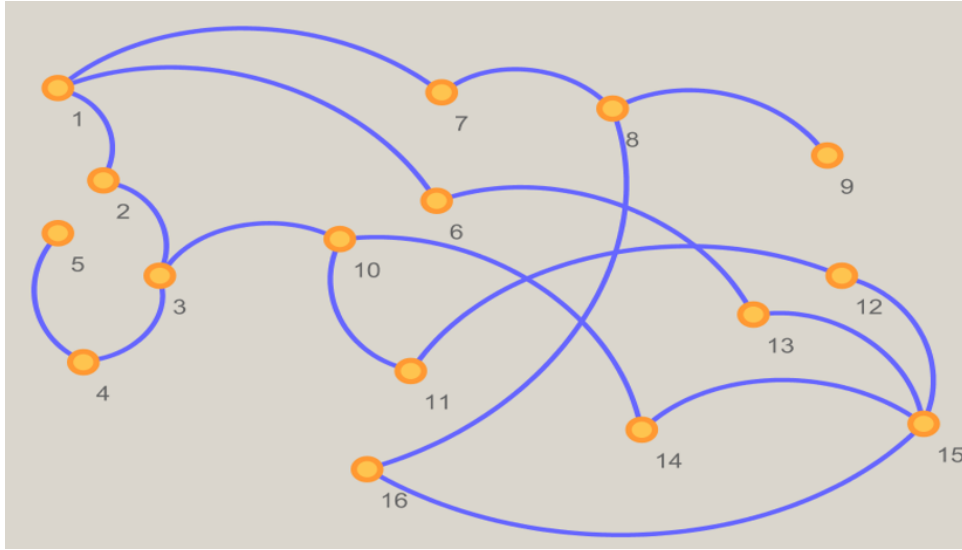


Figura 4.12: Red P2P no estructurada.

puede estar asociada con un conjunto de nodos en la red (aquellos que se encuentran procesando la consulta en el tiempo t). Se asume que el tiempo se incrementa en una unidad luego de que los nodos que se encuentran procesando mensajes en el tiempo t finalizan. A continuación se presenta una caracterización formal de la red P2P presentada en la figura 4.12 en su estado inicial, asumiendo $t = 0$ donde todos los nodos se encuentran en el estado ocioso.

$$P = (\mathcal{N}, \mathcal{E}, S_t)$$

$$\mathcal{N} = \{1, 2, \dots, 16\}$$

$$\mathcal{E} = \{(1,2), (1,7), (1,6), (2,3), (2,1), (3,2), (3,10), (3,4), (4,3), (4,5), (5,4), (6,1), (6,13), (7,1), (7,8), (8,7), (8,9), (9,8), (10,3), (10,14), (10,11), (11,10), (11,12), (12,11), (12,15), (13,6), (13,15), (14,10), (14,15), (15,13), (15,16), (15,12), (15, 14), (16,8), (16,15)\}$$

$$\forall n \in \mathcal{N}, S_0(n) = \text{ocioso}$$

4.2.2. El framework ArgP2P

Como se presentó anteriormente, una red P2P utiliza algoritmos con distintos grados de inteligencia para rutear consultas. Sin embargo, dichos algoritmos no están dotados de un mecanismo de inferencia poderoso dado que su lógica se limita al análisis básico de ciertas variables locales y estadísticas almacenadas en determinados nodos. Algunas

de las variables que se toman en cuenta al momento de determinar cómo proceder al recibir una consulta son la disponibilidad del nodo para responderla, la relevancia de la consulta respecto a los intereses del nodo, etc. Estas características son analizadas por el nodo en cuestión utilizando algoritmos imperativos, resultando de esta manera en un nodo con un comportamiento reactivo. En las búsquedas temáticas cada consulta Q se encuentra vinculada con un tópico T (por ejemplo, consulta = “números primos” y tópico = “matemática”) y cada nodo se encuentra interesado en un conjunto de *temáticas* (por ejemplo, el nodo 1 está interesado en el conjunto de temáticas {matemática, física}). Un nodo interesado en un tópico específico es un nodo candidato para responder consultas asociadas a ese tópico, además cada nodo de la red tiene la capacidad de adquirir conocimiento respecto a otros nodos y sus respectivos intereses. Este hecho motiva la siguiente definición:

Definición 2 Sea $P = (\mathcal{N}, \mathcal{E}, S_t)$ una red P2P. Para cada nodo $n \in \mathcal{N}$, se define una base de conocimiento dinámica KB_n con el siguiente formato:

$$KB_n = \langle (topico_1, \mathcal{N}_{topico_1}), (topico_2, \mathcal{N}_{topico_2}), \dots, (topico_n, \mathcal{N}_{topico_n}) \rangle,$$

donde \mathcal{N}_{topico_i} representa un conjunto de nodos asociados con $topico_i$.

El problema con este enfoque es que uno o más nodos pueden quedar desconectados de su comunidad, lo que se presentó en la sección anterior como el *Problema de las Comunidades Cerradas*. Como consecuencia de este problema puede suceder que algunos mensajes nunca encuentren una respuesta y que aquellos nodos que tienen menos recursos para compartir queden excluidos de sus comunidades. Para intentar atenuar este problema, se extendió el modelo anterior agregando una componente que permite tomar decisiones respecto a cuándo es un buen momento para explorar la red y así re-encontrar a aquellos nodos aislados con su comunidad. Esta nueva componente es llamada: *sistema argumentativo de toma de decisiones*.

Con el objetivo de capturar características importantes asociadas al comportamiento argumentativo, se definen formalmente los siguientes vectores booleanos:

Definición 3 Sea $P = (\mathcal{N}, \mathcal{E}, S_t)$ una red P2P. Para cada nodo $n \in \mathcal{N}$, una consulta entrante Q asociada con un tópico T , y su KB_n se define un vector de variables booleanas:

$$RB_n(Q, T) = \langle disponibilidad(Q, T), relevancia(Q, T), conocimiento(Q, T) \rangle,$$

donde:

- *disponibilidad(Q, T) es verdadero si y sólo si n (interesado en el tópico T) puede dar respuesta a la consulta entrante Q asociada al tópico T ;*
- *relevancia(Q, T) es verdadero si y sólo si el tópico T de la consulta entrante Q es relevante para el nodo n ;*
- *conocimiento(Q, T) es verdadero si y sólo si n tiene conocimiento respecto a otros nodos interesados en el tópico T de la consulta Q .*

$$AB_n(KB_n) = \langle \text{interes}(KB_n), \text{confiabilidad}(KB_n), \text{congestion}(KB_n), \text{necesidad_de_explorar}(KB_n) \rangle,$$

donde:

- *interes se encuentra garantizado si y sólo si el nodo n contiene información en su base de conocimiento KB_n sobre otros nodos y sus intereses;*
- *confiabilidad se encuentra garantizado si y sólo si el conocimiento almacenado en KB_n brinda razones para creer que ciertos nodos son confiables;*
- *congestion se encuentra garantizado si y sólo si el conocimiento almacenado en KB_n brinda razones para creer que existen caminos hacia ciertos nodos que no se encuentran congestionados;*
- *necesidad_de_explorar se encuentra garantizado si y sólo si el conocimiento almacenado KB_n brinda razones para creer que es necesario explorar la red.*

Definición 4 Sea $P = (\mathcal{N}, \mathcal{E}, S_t)$ una red P2P. Un Sistema Reactivo de Toma de Decisiones para un nodo $n \in \mathcal{N}$, denotado $RDMS_n$ (por sus siglas en inglés, *reactive decision-making system*), es un sistema imperativo que dada una consulta Q asociada a un tópico T , toma $RB_n(Q, T)$ y KB_n y retorna una decisión D indicando cómo rutear Q en P . Basándose en la consulta Q , su tópico T y en la base de conocimiento KB_n asociada al nodo n , $RDMS_n$ puede elegir uno de los siguientes cursos de acción:

- Descartar la consulta Q si su *TTL* ha expirado;
- Re-enviar la consulta Q hacia nodos interesados en el tópico T cuando se encuentre dicha información en la base de conocimiento KB_n ;

- Re-enviar la consulta Q hacia los nodos vecinos *cuando no cumplen las situaciones anteriores*.

Definición 5 Sea $P = (\mathcal{N}, \mathcal{E}, S_t)$ una red P2P. Un Sistema Argumentativo de Toma de Decisiones para un nodo $n \in \mathcal{N}$, denotado $ADMS_n$ (por sus siglas en inglés, *Argumentative Decision-Making System*) es un programa ABA (basado en KB_n , $AB_n(KB_n)$, Q y T) que determina si la decisión *necesidad_de_explorar*(KB_n) se encuentra garantizada bajo la semántica *grounded skeptical*. Si *necesidad_de_explorar*(KB_n) es verdadero, este sistema retorna verdadero y un algoritmo de propagación de consultas Flooding es utilizado para rutear la consulta Q . En otro caso el sistema retorna falso y la estrategia para el ruteo es la obtenida a partir de $RDMS_n$.

Definición 6 Sea $P = (\mathcal{N}, \mathcal{E}, S_t)$ una red P2P. Se define **RDMS** al conjunto de los sistemas $RDMS_n$ de todos los nodos $n \in \mathcal{N}$. Análogamente, se define **ADMS** como el conjunto de todos los sistemas $ADMS_n$ para todos los nodos $n \in \mathcal{N}$.

Definición 7 Sea $P = (\mathcal{N}, \mathcal{E}, S_t)$ una red P2P. Un framework ArgP2P es una terna $(P, \mathbf{RDMS}, \mathbf{ADMS})$ donde:

- P es una red P2P dinámica, $P = (\mathcal{N}, \mathcal{E}, S_t)$,
- **RDMS** representa a la componente del sistema reactivo de toma de decisiones de P , y
- **ADMS** representa a la componente del sistema argumentativo de toma de decisiones de P .

En la figura 4.13 se presenta una representación gráfica del framework *ArgP2P* para un nodo n . En este framework se combinan el *sistema reactivo de toma de decisiones* que utiliza el algoritmo 4.3.3 presentado en la sección anterior [NLMC17] y el *sistema argumentativo de toma de decisiones* [NMC15] (indicado con línea punteada en la figura). En el algoritmo 1 se presenta un código en alto nivel del algoritmo que corre cada nodo j (ejecutarNodo) en un framework *ArgP2P*. El algoritmo 2 muestra el pseudo-código del *sistema reactivo de toma de decisiones* y finalmente el algoritmo 3 muestra el comportamiento a alto nivel del *sistema argumentativo de toma de decisiones*. El esquema

Algoritmo 1: Algoritmo ejecutarNodo para cada nodo j en un framework $ArgP2P$.

Input: Una consulta Q asociada a un t3pico T .

Output: Una decisi3n D respecto a qu3 acci3n tomar luego del arribo de la consulta Q .

$count := count + 1$; $\backslash \backslash$ count es una variable global asociada a cada nodo, inicialmente establecida en 0, e incrementada cada vez que el algoritmo ejecutarNodo es ejecutado.

CargarBaseDeConocimiento(KB_j); $\backslash \backslash$ Inicializa la base de conocimiento con el conocimiento que tiene el nodo.

Decision $D :=$ SistemaReactivoDeTomaDeDecisiones(RB_j, Q, T, KB_j);

if ($ttarg = 0$) **then**

if ($SistemaArgumentativoDeTomaDeDecisiones(AB_j, KB_j)$) **then**

$D = enviar_a_Nodos_Vecinos(Q)$;

end

end

return D .

Algoritmo 2: Sistema reactivo de toma de decisiones para cada nodo j ($RDMS_j$)

Input: Una consulta Q asociada a un t3pico T , un vector de variables RB_j y una base de conocimiento KB_j .

Output: Una decisi3n D respecto a qu3 acci3n tomar luego del arribo de la consulta Q .

for each $p_i \in RB_j$ **do**

 verificar(p_i, KB_j);

end

$D :=$ tomarDecision ($Q, T, RB_j(Q, T), KB_j$);

return D .

presentado en la figura anterior junto con los algoritmos representan los aspectos m3s sobresalientes del framework $ArgP2P$.

El algoritmo principal muestra la interacci3n entre los dos sistemas que componen el framework $ArgP2P$: El *sistema reactivo de toma de decisiones* ($RDMS_j$) y el *sistema argumentativo de toma de decisiones* ($ADMS_j$) en un determinado nodo j miembro de la red. La argumentaci3n es un proceso computacionalmente costoso y dado que el objetivo

Algoritmo 3: Sistema argumentativo de toma de decisiones para cada nodo j ($ADMS_j$)

Input: Una base de conocimiento KB_j y un vector de variables $AB_j(KB_j)$.

Output: Retorna verdadero si y sólo si $necesidad_de_explorar(KB_j)$ está garantizado bajo la semántica grounded skeptical, en otro caso retorna falso.

boolean *resultado*;

construirProgramaABA($KB_j, AB_j(KB_j)$);

resultado := ejecutarProgramaABA($necesidad_de_explorar(KB_j)$);

return *resultado*.

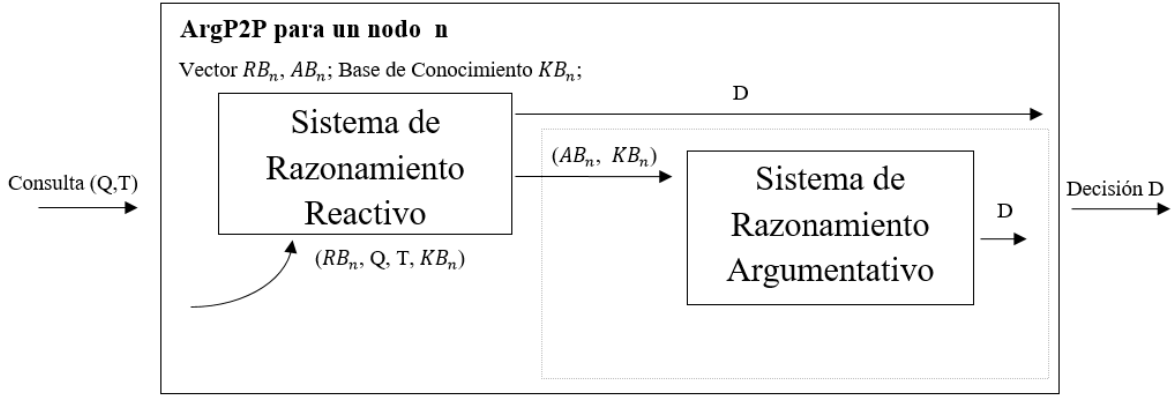


Figura 4.13: Descripción a alto nivel de las componentes de cada nodo n de un framework.

de *ArgP2P* es optimizar el proceso de búsqueda disminuyendo los tiempos de respuestas, es necesario limitar la ejecución del sistema argumentativo. Para sobrellevar esta cuestión el framework cuenta con el parámetro ajustable *ttarg* el cuál permite determinar qué tan reactivo o argumentativo son los nodos. Desde el algoritmo principal se puede apreciar que el $ADMS_j$ se ejecutará con una frecuencia determinada por el parámetro *ttarg*. El $RDMS_j$ verifica las variables del vector RB_j usando el conocimiento almacenado en su KB_j para tomar una decisión respecto a cómo rutear la consulta Q . El $ADMS_j$ construye un programa ABA a partir del vector AB_j y la base de conocimiento KB_j . Luego ejecuta el programa ABA para llegar a la decisión final respecto a si es necesario o no explorar la red con el objetivo de aumentar el conocimiento global.

4.2.3. Caso de estudio

En esta sub-sección se presenta un caso de estudio simple, cuyo objetivo es ilustrar mediante un ejemplo significativo cuál es el aporte del sistema argumentativo al framework *ArgP2P*. Al final de esta sub-sección se muestra uno de los programas lógicos creados por el framework y cómo a partir del mismo se deriva acertadamente la decisión de explorar la red.

Para este caso de estudio se considera la red P2P que se muestra en la figura 4.12, además se supone que los nodos 1, 2, 8 y 10 están interesados en el tópico *matemática* y tienen el conocimiento que se muestra en la tabla 4.2 respecto al resto de la red.

KB_n	Nodos asociados al tópico “matemática”		
KB_1	5	8	10
KB_5	1	8	10
KB_8	1	5	10
KB_{10}	1	5	8

Tabla 4.2: Conocimiento de los nodos respecto al tópico “matemática”.

Supongamos que el nodo 1 genera un mensaje de consulta. Para propagar este mensaje verificará su base de conocimiento y de acuerdo a ello lo enviará a aquellos nodos que tengan interés en esa consulta. La figura 4.14(a) muestra los caminos que inicialmente toman estos mensajes que se encuentran representados mediante flechas punteadas. Teniendo en cuenta la definición presentada en 4.2.1 la representación formal de la red en un tiempo ($t = 1$) es la siguiente:

$$\begin{aligned}
 S_1(n) &= \text{ocioso} \quad \forall n \in \mathcal{N} - \{1\}, \\
 S_1(1) &= (\text{consultar}, \{5, 8, 10\}).
 \end{aligned}$$

Luego el nodo 5 recibe el mensaje re-enviado y no lo puede responder, por lo tanto lo re-envía a los nodos potencialmente útiles que conoce mediante su KB_5 (excepto al nodo 1 que fue quien originó la consulta); la figura 4.14(b) ilustra esta situación. Los nodos 8 y 10 tampoco pueden contestar la consulta así que la re-envían teniendo en cuenta sus bases de conocimiento KB_8 y KB_{10} respectivamente. La figuras 4.14(c) y 4.14(d) ilustran este escenario. Todas estas acciones se llevan a cabo en el momento de tiempo $t = 2$ y su representación formal es:

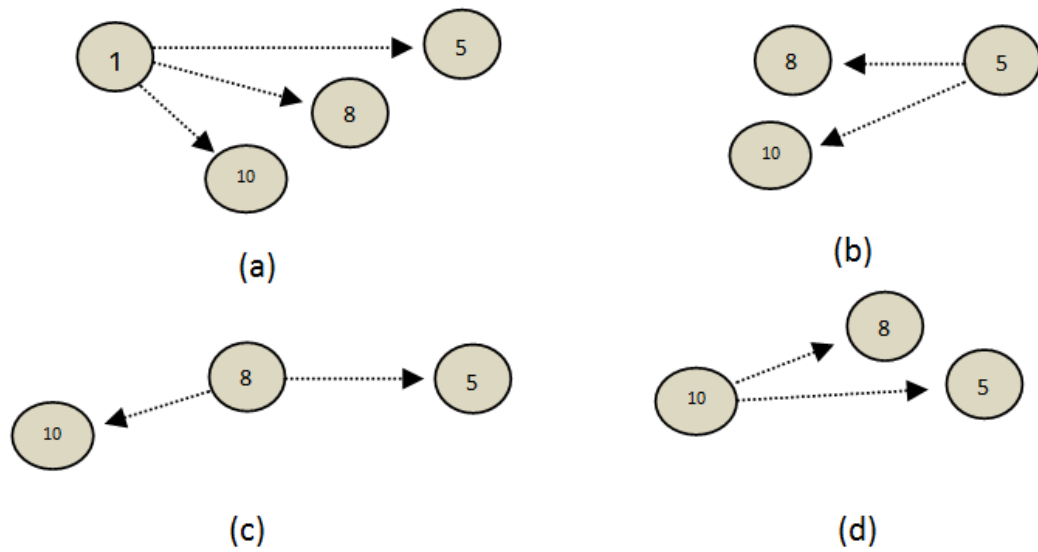


Figura 4.14: Representación gráfica del envío de mensajes durante el ruteo de consultas en una red P2P (caso de estudio).

$$\begin{aligned}
S_2(n) &= \text{ocioso} \quad \forall n \in \mathcal{N} - \{5, 8, 10\}, \\
S_2(5) &= (\text{reenviar}, \{8, 10\}), \\
S_2(8) &= (\text{reenviar}, \{5, 10\}), \\
S_2(10) &= (\text{reenviar}, \{5, 8\}).
\end{aligned}$$

Luego de estos re-envíos, el nodo 5 recibe un mensaje desde el nodo 8 que fue generado por el nodo 1 y un mensaje desde el nodo 10 que también fue originado por el nodo 1; finalmente el nodo 10 recibe un mensaje desde el nodo 5 generado por el nodo 1 y otro desde el nodo 8 también con origen en el nodo 1. Todos estos mensajes son descartados en un tiempo $t = 3$ dado que los nodos receptores ya tienen en sus registros información de haber procesado un mensaje con ese mismo identificador (id). Como resultado, las acciones que se ejecutan en el tiempo $t = 3$ son las siguientes:

$$\begin{aligned}
S_3(n) &= \text{ocioso} \quad \forall n \in \mathcal{N} - \{5, 8, 10\}, \\
S_3(5) &= \text{descartar}, \\
S_3(8) &= \text{descartar}, \\
S_3(10) &= \text{descartar}.
\end{aligned}$$

En este punto es donde se evidenció el *Problema de las Comunidades Cerradas*, la consulta circuló dentro de la comunidad y dado que ningún miembro pudo dar respuesta terminó siendo descartada. Ahora supongamos que para que esta situación no ocurra el nodo 8 decide explorar la red. Bajo este nuevo escenario en el momento $t = 3$ re-enviaría la consulta hacia los nodos 7, 9 y 16. El nodo 7 sólo tiene como vecino al nodo 1 que fue quién originó la consulta, por este motivo no es considerado. El nodo 9 no tiene vecinos para hacer el re-envío y tiene como vecino al nodo 8 que fue desde quien arribó el mensaje, por este motivo no es considerado. Luego, en el momento $t = 4$, el nodo 16 re-enviará la consulta al nodo 15 quien finalmente en el momento de tiempo $t = 5$ está en condiciones de responder a la consulta originada por el nodo 1.

Formalmente:

$$\begin{aligned}
S_3(n) &= \text{ocioso} \quad \forall n \in \mathcal{N} - \{5, 8, 10\}, \\
S_3(5) &= \text{descartar}, \\
S_3(8) &= (\text{reenviar}, \{7, 9, 16\}), \\
S_3(10) &= \text{descartar}.
\end{aligned}$$

$$\begin{aligned}
S_4(n) &= \text{ocioso} \ \forall \ n \in \mathcal{N} - \{7, 9, 16\}, \\
S_4(7) &= \text{descartar}, \\
S_4(9) &= \text{descartar}, \\
S_4(16) &= (\text{reenviar}, \{15\}).
\end{aligned}$$

$$\begin{aligned}
S_5(n) &= \text{ocioso} \ \forall \ n \in \mathcal{N} - \{15\}, \\
S_5(15) &= (\text{responder}, 1).
\end{aligned}$$

Cuando el $ADMS_8$ se ejecuta, un programa ABA es construido. Las suposiciones, reglas y contras que constituyen el programa pueden ser automáticamente generadas a partir del conocimiento que tiene un nodo a partir del vector AB_8 y de la base de conocimiento KB_8 . La ejecución de este programa le permite al nodo 8 determinar si es conveniente o no la exploración. Si bien la respuesta es simplemente un valor de verdad, su impacto puede ser muy fuerte en todo el sistema. En este ejemplo que se desarrolla como caso de estudio, el hecho de decidir no explorar significa no encontrar respuesta a la consulta. Si se toma en cuenta un escenario opuesto en el que un nodo decide explorar y aún no era necesario, se estaría aumentando el tiempo de respuesta y sobrecargando la red con mensajes innecesarios. Otro aspecto a tener en cuenta es que se está manejando información potencialmente inconsistente, problema que es resuelto sin mayores complicaciones por el razonamiento argumentativo. Cuando el nodo decide explorar, descarta la decisión tomada inicialmente por $RDMS_8$. A continuación se muestra un programa lógico generado por $ADMS_8$ junto al árbol de derivación que muestra cómo la decisión de explorar es alcanzada. El árbol es presentado en la figura 4.15.

$A = \{\text{noCongestionado}(1), \text{noCongestionado}(5), \text{noCongestionado}(10),$
 $\text{buenCandidato}(1), \text{buenCandidato}(5), \text{buenCandidato}(10), \text{siempreCongestionado}(1),$
 $\text{nuncaCongestionado}(5), \text{nuncaCongestionado}(10), \text{largoTiempoParaResponder}(1),$
 $\text{necesidad_de_explorar}\}$

$R = \{\text{matematica}(1) \leftarrow;$
 $\text{matematica}(5) \leftarrow; \text{matematica}(10) \leftarrow; \text{buenTiempoDeRespuesta}(1) \leftarrow;$
 $\text{buenTiempoDeRespuesta}(5) \leftarrow;$
 $\text{reenviar}(1) \leftarrow \text{matematica}(1), \text{noCongesitonado}(1), \text{confiable}(1);$
 $\text{reenviar}(5) \leftarrow \text{matematica}(5), \text{noCongesitonado}(5), \text{confiable}(5);$

```

reenviar(10) ← matematica(10), noCongestionado(10), confiable(10);
congestionado(1) ← largoTiempoEnResponder(1), siempreCongestionado(1);
congestionado(5) ← largoTiempoEnResponder(5), siempreCongestionado(5);
congestionado(10) ← largoTiempoEnResponder(10), siempreCongestionado(10);
confiable(1) ← buenTiempoDeRespuesta(1), buenCandidato(1);
confiable(5) ← buenTiempoDeRespuesta(5), buenCandidato(5);
confiable(10) ← buenTiempoDeRespuesta(10), buenCandidato(10);
noExplorar ← reenviar(1);
noExplorar ← reenviar(5);
noExplorar ← reenviar(10)}

```

```

 $\overline{\text{noCongestionado}(1)} = \text{congestionado}(1);$ 
 $\overline{\text{noCongestionado}(5)} = \text{congestionado}(5);$ 
 $\overline{\text{noCongestionado}(10)} = \text{congestionado}(10);$ 
 $\overline{\text{buenCandidato}(1)} = \text{malCandidato}(1);$ 
 $\overline{\text{buenCandidato}(5)} = \text{malCandidato}(5);$ 
 $\overline{\text{buenCandidato}(10)} = \text{malCandidato}(10);$ 
 $\overline{\text{siempreCongestionado}(1)} = \text{nuncaCongestionado}(1);$ 
 $\overline{\text{nuncaCongestionado}(5)} = \text{siempreCongestionado}(5);$ 
 $\overline{\text{nuncaCongestionado}(10)} = \text{siempreCongestionado}(10);$ 
 $\overline{\text{largoTiempoEnResponder}(1)} = \text{bajoTiempoEnResponder}(1);$ 
 $\overline{\text{necesidad\_de\_explorar}} = \text{reenviar}(1);$ 
 $\overline{\text{necesidad\_de\_explorar}} = \text{reenviar}(5);$ 
 $\overline{\text{necesidad\_de\_explorar}} = \text{reenviar}(10);$ 
 $\overline{\text{necesidad\_de\_explorar}} = \text{noExplorar};$ 

```

En el programa de ejemplo presentado anteriormente se puede visualizar el manejo de inconsistencias: la decisión $\overline{\text{necesidad_de_explorar}}$ es inconsistente con $\text{reenviar}(1)$, de la misma manera la presunción $\overline{\text{noCongestionado}(1)}$ es inconsistente con la conclusión $\text{congestionado}(1)$.

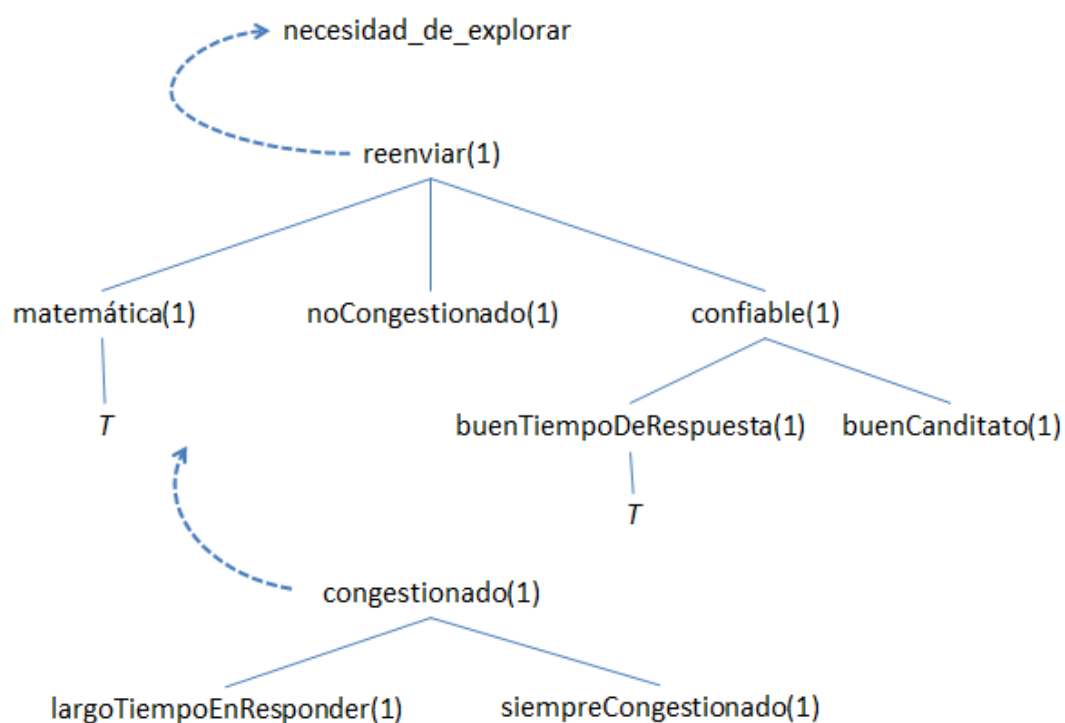


Figura 4.15: Argumentos y ataques involucrados en el árbol de derivación que soporta la decisión “*necesidad_de_explorar*”. Las flechas punteadas representan ataques entre argumentos.

4.3. Resumen del capítulo

En este capítulo, en primera instancia, se presentaron una serie de algoritmos que explotan el hecho de que si se dirigen las búsquedas según los intereses de los participantes, emergerán patrones de comunicación eficientes que llevan directamente a la mejora en el rendimiento del sistema. Cada uno de estos algoritmos fue presentado junto a un ejemplo ilustrativo de su comportamiento ante situaciones particulares. Además se presentó una tabla en la que se comparan los aspectos en común entre todos ellos. Una de las limitaciones observadas es lo que se describió como *Problema de las Comunidades Cerradas*. Bajo este escenario uno o más nodos pueden quedar separados de su comunidad, o hasta incluso puede existir una misma comunidad dividida en dos o más comunidades pequeñas. La emergencia de este tipo de comunidades desconectadas es perjudicial para la interacción de los participantes bajo un sistema de búsqueda temática, por lo tanto es un aspecto sobresaliente el hecho de encontrar algún mecanismo para atenuar los efectos de este problema. Para ayudar a minimizar el impacto del *Problema de las Comunidades Cerradas* se creó el framework *ArgP2P*, un novedoso modelo de búsqueda temática en redes P2P donde cada nodo de la red tiene la habilidad de combinar un comportamiento reactivo con un comportamiento racional (argumentativo). La máquina de inferencia argumentativa se provee mediante un programa ABA que posibilita alcanzar decisiones más acertadas respecto a qué pasos seguir tras la llegada de un mensaje de consulta a un nodo. Este framework brinda un parámetro ajustable denominado *ttarg* que permite hacer a los participantes más “reactivos” o más “racionales” dependiendo de qué tan frecuentemente se ejecute el sistema argumentativo. Para concluir el capítulo se presentó un conciso caso de estudio en donde se ilustra de manera sencilla el *Problema de las Comunidades Cerradas* y como *ArgP2P* puede minimizar su impacto gracias a su poderoso sistema argumentativo.

Capítulo 5

Evaluación

Como se ha mencionado a lo largo de esta Tesis, Internet cumple un rol fundamental en lo que a intercomunicaciones respecta. Las redes P2P son un modelo que utiliza protocolos específicos de comunicación para efectuar estas intercomunicaciones de manera eficiente. Los algoritmos presentados como contribución principal junto con *ArgP2P* pueden ser adoptados como parte de este protocolo. Para demostrar su eficiencia es importante llevar a cabo una evaluación empírica junto con un avanzado análisis de los datos obtenidos. Este capítulo se encuentra organizado de la siguiente manera: en la sección 5.1 se muestra cómo se evaluó cada uno de los algoritmos inteligentes junto con un análisis de los resultados obtenidos mediante las simulaciones. En la sección 5.2 se muestra cómo se evaluó el sistema argumentativo de *ArgP2P*. Finalmente, en la sección 5.3, se presenta una evaluación integral del framework *ArgP2P* la cual incluye simulaciones realistas, junto con los datos obtenidos a partir de ellas que sustentan la hipótesis de que dotando al sistema con racionalidad se pueden mejorar los patrones de comunicación. Con el objeto de mostrar el potencial que ofrece *ArgP2P*, se analizan las mejoras alcanzadas al intentar aplicar dicho framework para limitar las consecuencias del *Problema de las Comunidades Cerradas*. Para concluir este capítulo se presenta un resumen del mismo en la sección 5.4.

5.1. Evaluación de los algoritmos inteligentes

En esta sección se muestra cómo se llevaron a cabo las simulaciones de cada uno de los algoritmos inteligentes presentados previamente en el capítulo 4. La sección se encuentra

organizada como se detalla a continuación. En la sub-sección 5.1.1 se detalla cómo se efectuaron las simulaciones. En la sub-sección 5.1.2 se muestra un análisis de los datos obtenidos tras las simulaciones donde se evidencia la presencia de patrones eficientes de comunicación. En la sub-sección 5.1.3 se muestra una evaluación respecto a la escalabilidad de uno de los algoritmos que ha tenido mejor desempeño. Para completar este análisis se llevaron a cabo estudios de los datos respecto a su *Distribución de Grados*, resultados presentados en la sub-sección 5.1.4. Para finalizar, en la sub-sección 5.1.5 se presenta una evaluación de la topología subyacente de las comunidades semánticas que emergen gracias a la eficiencia lograda en las comunicaciones.

5.1.1. Configuraciones de la simulación

Un gran número de simulaciones se llevaron a cabo utilizando datos realistas con el objetivo de comparar el rendimiento de los algoritmos propuestos. Las simulaciones se efectuaron sobre dos redes físicas: una con 1000 nodos y otra con 5000. Estas redes fueron generadas utilizando el modelo Erdős-Rényi [ER59]. El parámetro p de este modelo fue establecido de modo tal de que cada nodo tenga un grado promedio de aproximadamente 10 (para $|N| = 5000$, $p = 0.002$). Para asegurar el correcto funcionamiento de los algoritmos se corroboró que ambas redes físicas se encontraran totalmente conectadas. En este punto se hace indispensable tener en claro la diferencia entre una *red física* y una *red lógica*. La red física es aquella que representa a las componentes físicas del problema (los dispositivos y sus conexiones) y permanece estática durante las simulaciones. Por otro lado, la red lógica es una red que emerge como resultado de la interacción entre los miembros de la red. En el caso de nuestros algoritmos la topología de esta red cambia a medida que se incrementa el aprendizaje global de todo el sistema. Durante las simulaciones no se utilizaron índices ni directorios centralizados. Cada uno de los nodos de la red es tanto un proveedor como un demandante de información en el sistema. Además cada uno de ellos tiene asociado uno o más tópicos de interés.

Uno de los primeros desafíos encontrados para llevar a cabo las simulaciones fue seleccionar un framework apropiado. Se estudiaron las distintas alternativas y se optó por el framework OmNet++ [Pon93, VH08] dada su flexibilidad, facilidad de uso y sus frecuentes publicaciones y actualizaciones. Para llevar a cabo las visualizaciones de las redes lógicas se utilizó JUNG (Java Universal Network/Graph Framework) [jun] y la herra-

mienta LaNet-vi tool¹ (Large Networks Visualization Tool). Esta última permite obtener visualizaciones representativas de las descomposiciones k -núcleos (en inglés, k -core) de las redes lógicas. Como entrada de las simulaciones se utilizaron 45,742 artículos científicos obtenidos de la biblioteca científica SciELO². Los artículos utilizados están clasificados en 12 tópicos. Dicha clasificación es la que brinda SciELO y la cantidad de artículos correspondientes a cada tópico se muestra en la tabla 5.1. Estos tópicos fueron distribuidos entre los nodos de modo tal que cada uno de los nodos esté interesado en uno o más tópicos. Los tópicos de aquellos nodos que tienen más de un interés se encuentran relacionados entre sí. Por ejemplo, un determinado nodo puede estar interesado en ingeniería y matemática pero no en humanidades e ingeniería. Como último paso para terminar la configuración se repartieron los artículos aleatoriamente entre los nodos según las temáticas de interés de cada uno, de esta manera cada nodo contiene sólo artículos de su interés.

Tópico	Número de artículos
T1: Ciencias Agrícolas	2,403
T2: Ciencias Biológicas	4,250
T3: Ciencias de la Salud	23,754
T4: Ciencias de la Tierra	2,775
T5: Ciencias Geológicas	197
T6: Ciencias Sociales	1,705
T7: Ciencias Sociales Aplicadas	4,421
T8: Humanidades	4,652
T9: Ingeniería	1,289
T10: Lingüística, Letras y Arte	164
T11: Matemática	9
T12: Química	123
Número total de artículos:	45,742

Tabla 5.1: Número de artículos asociados a cada tópico.

Con el propósito de comparar los algoritmos propuestos para el ruteo de consultas, cada algoritmo fue inicialmente ejecutado diez veces con una red física de 1000 nodos. Durante cada ejecución, las consultas fueron iniciadas por nodos seleccionados de manera aleatoria. El número de consultas en cada ejecución fue de 150 y la máxima cantidad de saltos antes de descartar la consulta fue establecida en 50. Cada ejecución nueva del mismo algoritmo se ve beneficiada por el aprendizaje obtenido en las ejecuciones anteriores. Se utilizaron los siguientes criterios de comparación:

¹<http://lanet-vi.fi.uba.ar/>

²<http://www.scielo.org>

- El coeficiente de clustering promedio de la red lógica.
- El número de consultas que encontraron su respuesta.
- El número de mensajes enviados por cada nodo, teniendo en cuenta los mensajes de actualización para analizar si este tipo de mensajes se encuentra congestionando la red.
- La cantidad de saltos necesarios para alcanzar una respuesta.

Todas las simulaciones fueron ejecutadas en un servidor con las siguientes características:

- 32 procesadores (4 x 8 cores) Opteron.
- 128 GB RAM.
- Debian GNU/Linux 6.0 64 bits.
- kernel 3.8.3.
- Oracle JRE 1.7.0 21.

5.1.2. Surgimiento de patrones de comunicación y estructuras con topología de mundo pequeño

En un sistema P2P el objetivo principal es obtener información requerida por los usuarios en un tiempo aceptable, es por ello que es sumamente importante evaluar la cantidad de consultas satisfechas. La tabla 5.2 muestra el rendimiento de los algoritmos evaluados respecto a este aspecto. Los resultados se corresponden a las ejecuciones número uno, tres, cuatro, siete y diez de cada uno de los algoritmos. Los resultados indican que el número de consultas satisfechas es alto durante la primera ejecución. Esto se debe a que durante dicha ejecución el conocimiento de los nodos respecto al resto de la red es limitado y la mayoría de los mensajes son propagados por toda la red, de este modo hay más probabilidades de encontrar una respuesta con la contrapartida de sobrecargar la red con mensajes replicados. En el resto de las simulaciones el número de respuestas encontradas tiende a disminuir levemente o a mantenerse constante. Aunque el número

Consultas satisfechas									
Ejecución	Algoritmo								
	1.0	2.0	2.1	3.0	3.1	3.2	3.3	4.3.3	5.0
1	92	142	125	115	100	106	110	135	128
3	80	81	98	99	94	98	97	97	94
5	63	87	92	90	93	92	93	94	89
7	120	84	97	88	102	100	86	83	95
10	75	80	94	92	104	97	92	91	98

Tabla 5.2: Tabla comparativa del número de respuestas satisfechas ($|N| = 1000$).

de respuestas sea un poco menor, la cantidad de saltos requeridos para encontrarlas es sumamente bajo.

Tal como se mencionaba anteriormente, el tiempo de respuesta es sumamente importante en un sistema donde el usuario requiere información y espera una respuesta en un tiempo tolerable. Por este motivo se evaluó cuidadosamente el número de saltos en promedio que necesita una consulta para encontrar su respuesta. Los resultados se muestran en la tabla 5.3 y en el gráfico 5.1. A partir de estos resultados se puede concluir que el número de saltos decrece a medida que el conocimiento global de la red aumenta. Otro aspecto que se puede apreciar es que las versiones 4.3.3 y 5.0 exhiben una reducción del número de saltos mucho más significativo que el resto de los algoritmos. Esto se debe a que estos dos algoritmos son los más evolucionados respecto a la forma en la que incorporan y comparten su conocimiento.

La comunidad científica ha desarrollado distintas métricas para medir el nivel de aprendizaje. Estas métricas ofrecen diferentes ventajas, dependiendo del entorno de aprendizaje donde se lleva a cabo el análisis. En el caso de los sistemas P2P se ha evidenciado que a medida que los nodos incorporan aprendizaje tienden a comunicarse con un conjunto de nodos específicos. En nuestro caso estos conjuntos de nodos comparten intereses en común. Todo esto lleva a que el coeficiente de clustering sea una métrica apropiada para analizar y comparar el aprendizaje global logrado entre los algoritmos propuestos. Un coeficiente de clustering alto y un diámetro pequeño indican la presencia de *comunidades semánticas* en la red [WS98]. La tabla 5.4 y la figura 5.2 muestran que el coeficiente de clustering promedio se incrementa considerablemente a medida que las ejecuciones avan-

Ejecución	Saltos								
	Algoritmo								
	1.0	2.0	2.1	3.0	3.1	3.2	3.3	4.3.3	5.0
1	30	37	49	48	46	46	45	25	29
3	29	48	39	27	6	32	8	2	2
5	29	46	20	12	5	8	26	2	2
7	28	49	18	7	5	17	3	2	2
10	30	50	21	7	5	10	9	2	2

Tabla 5.3: Tabla comparativa del número de saltos para encontrar una respuesta ($|N| = 1000$).

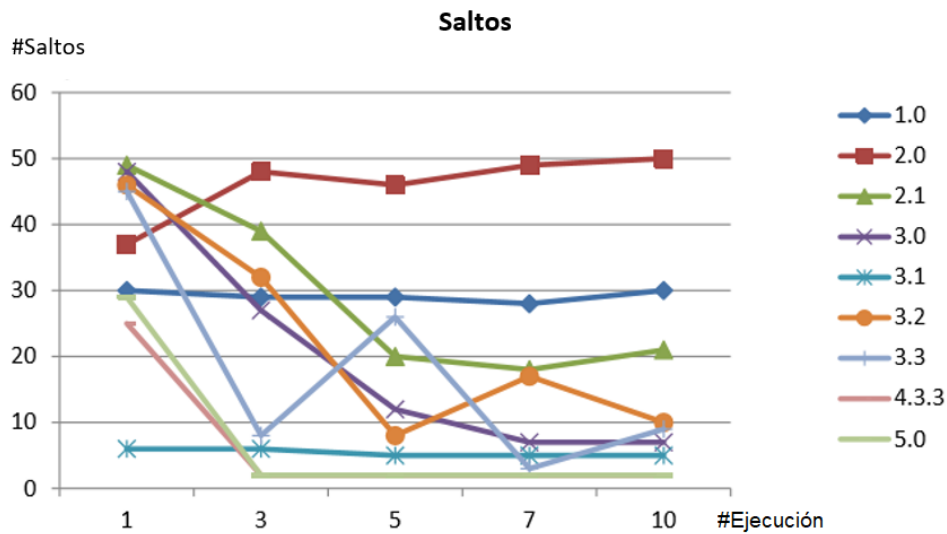


Figura 5.1: Figura comparativa del número de saltos para encontrar una respuesta ($|N| = 1000$).

zan. Este incremento se debe a que durante las últimas corridas el aprendizaje global de la red es más alto y los nodos pueden enviar las consultas directamente hacia los nodos potencialmente útiles.

Otro de los aspectos que no hay que pasar por alto es la cantidad de mensajes que fueron enviados en toda la red. Se sabe que los algoritmos que incorporan los mensajes de actualización podrían llegar a sobrecargar la red. Los resultados obtenidos se muestran en la tabla 5.5 y en la figura 5.3. Como se puede desprender de la tabla, en todos los

Coeficiente de clustering									
Ejecución	Algoritmo								
	1.0	2.0	2.1	3.0	3.1	3.2	3.3	4.3.3	5.0
1	0	0.0984	0.1016	0.1955	0.2713	0.2921	0.3638	0.696	0.6862
3	0	0.1510	0.1548	0.2450	0.2754	0.3286	0.4806	0.7062	0.7013
5	0	0.1787	0.1676	0.2443	0.2788	0.3356	0.5146	0.7088	0.7052
7	0	0.1914	0.1826	0.2551	0.2805	0.3380	0.5228	0.7098	0.7093
10	0	0.2130	0.1906	0.2636	0.2827	0.3434	0.5274	0.7135	0.7099

Tabla 5.4: Tabla comparativa del promedio de coeficiente de clustering ($|N| = 1000$).

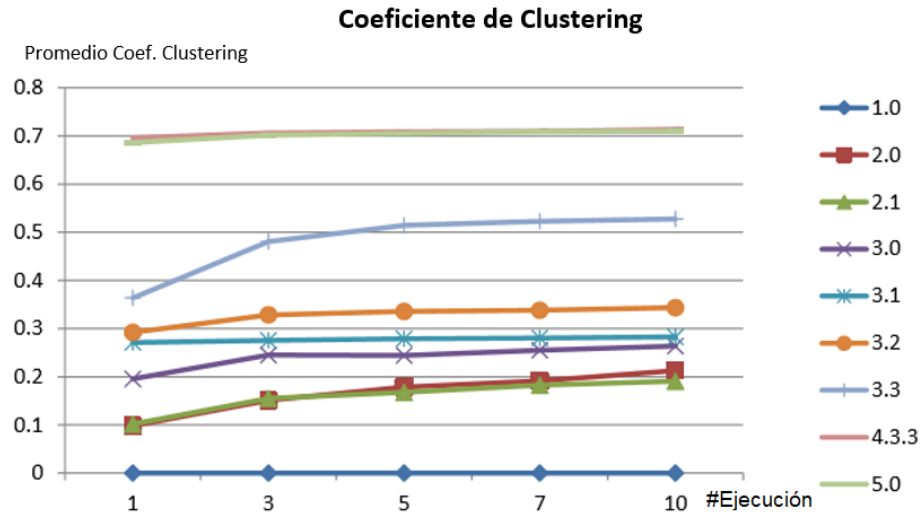


Figura 5.2: Figura comparativa del promedio de coeficiente de clustering ($|N| = 1000$).

casos la cantidad de mensajes tiende a decrecer aún en aquellos algoritmos que utilizan los mensajes de actualización. Esto se debe a que este tipo de mensajes se usa racionalmente para compartir conocimiento que luego facilita el ruteo de las consultas hacia nodos potencialmente útiles.

Siguiendo el análisis respecto a los mensajes de actualización, la tabla 5.6 junto con la figura 5.4 muestran resultados que indican que el algoritmo 5.0 envía una menor cantidad de mensajes de actualización que el algoritmo 4.3.3 sin modificar el aprendizaje global de la red. Los algoritmos 2.0 y 2.1 no utilizan este tipo de mensajes mientras que el resto de los algoritmos muestra un número pequeño de mensajes de actualización debido a que no

Mensajes enviados									
Ejecución	Algoritmo								
	1.0	2.0	2.1	3.0	3.1	3.2	3.3	4.3.3	5.0
1	996415	108562	110423	109723	101479	108955	439165	1964130	1898622
3	998618	77250	77100	75186	93264	54257	421828	1462712	1447162
5	998630	64512	64659	64817	95536	71238	373371	1367388	1409499
7	998609	55360	57023	55128	98048	80699	267511	1345714	1391244
10	995254	41250	46263	42543	83674	90500	114423	1239345	1186936

Tabla 5.5: Tabla comparativa del número de mensajes enviados ($|N| = 1000$).

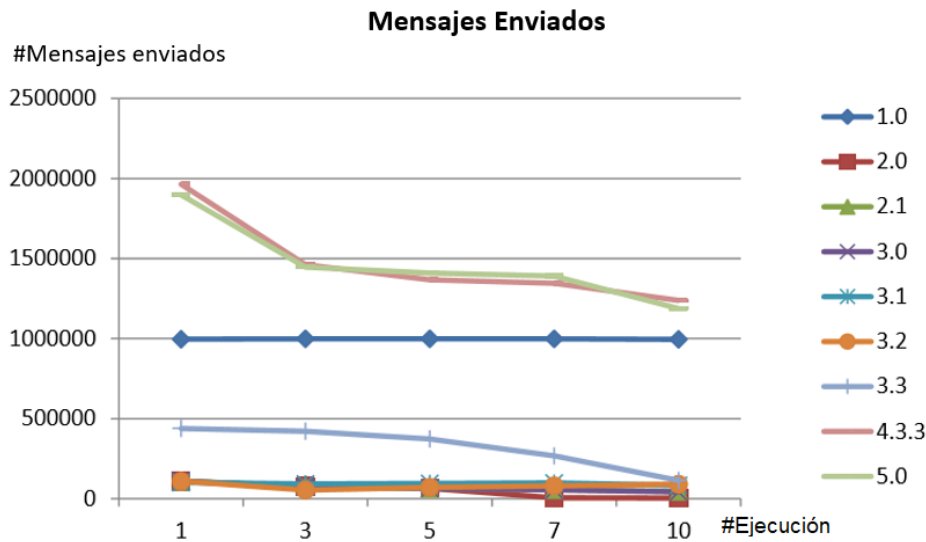


Figura 5.3: Figura comparativa del número de mensajes enviados ($|N| = 1000$).

comparten el conocimiento con sus comunidades.

Un detalle importante que se desprende del análisis de los datos presentados es que existe una relación entre la cantidad de mensajes de actualización enviados y el grado en el que la red alcanza un aprendizaje alto. Se evidencia que si bien los algoritmos más prometedores utilizan más mensajes de actualización, alcanzan un grado alto de aprendizaje más rápidamente, lo que evita la réplica de mensajes de consulta.

Mensajes de actualización						
Ejecución	Algoritmo					
	3.0	3.1	3.2	3.3	4.3.3	5.0
1	10755	98300	21881	62152	214890	201085
3	2215	2340	7620	60448	141792	126007
5	2172	2340	4991	61062	129781	144702
7	2612	2340	4929	45690	128532	119548
10	3789	2340	4046	18345	127907	114369

Tabla 5.6: Tabla comparativa del número de mensajes de actualización ($|N| = 1000$).

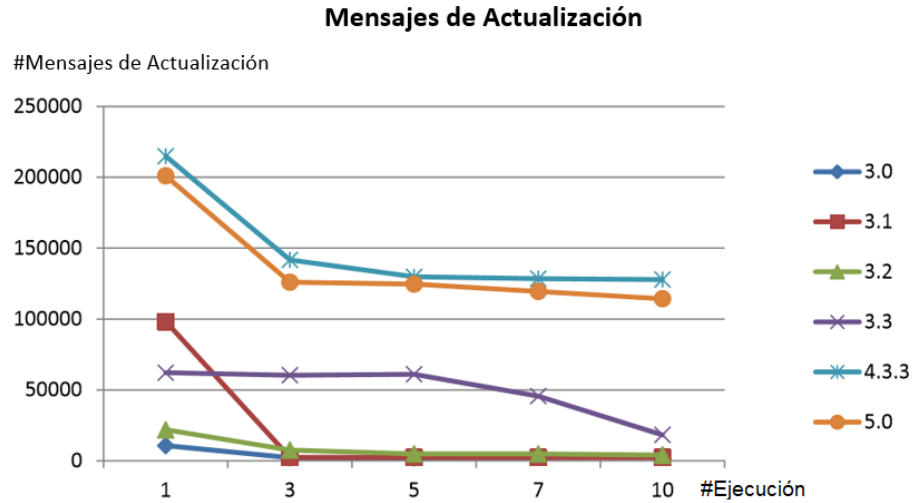


Figura 5.4: Gráfico comparativo del número de mensajes de actualización ($|N| = 1000$).

5.1.3. Evaluación de la escalabilidad

Como ocurre con la mayoría de los sistemas a gran escala, la escalabilidad es una cuestión de suma importancia. Las redes P2P son típicamente escalables, lo que le brinda a la red flexibilidad y mayor capacidad de almacenamiento. Más allá de esto, la verificación de la escalabilidad de un sistema distribuido, utilizando un framework que corre en una arquitectura centralizada, se ve afectada por limitaciones de hardware producto de usar un único servidor. Si bien la escalabilidad no pudo ser analizada de manera realista, se evaluó el algoritmo 4.3.3 con una red de 5000 nodos con el objetivo de verificar si el rendimiento era similar al obtenido por el mismo algoritmo con una red de 1000 participantes. Bajo las

simulaciones efectuadas con la red más grande, se realizaron 750 consultas inicializadas por nodos aleatorios tal como se realizó en las simulaciones anteriores y la cantidad de saltos para descartar un mensaje se estableció en 15. Los resultados de estas simulaciones se presentan en la tabla 5.7. Según las métricas analizadas se puede evidenciar que los patrones de conectividad y de comunicaciones mantienen comportamientos similares en una red de 1000 nodos respecto a otra red de 5000 nodos.

Ejecución	Número de saltos	Coefficiente de Clustering	Consultas Satisfechas	Mensajes enviados	Mensajes de actualización
1	10	0.0913	53	9806320	1052825
3	2	0.2410	234	7257560	709777
5	1	0.2664	401	6925420	649921
7	1	0.2755	538	6825532	648989
10	1	0.2810	567	6196961	635841

Tabla 5.7: Análisis del rendimiento del algoritmo 4.3.3 ($N = 5000$ nodes).

5.1.4. Distribución de grados y descomposición en k -núcleos

Anteriormente analizamos en detalle el coeficiente de clustering y el diámetro (basado en cantidad de saltos que demora encontrar una respuesta) obtenidos en cada una de las simulaciones, lo que demostraba que los patrones de comunicación establecidos daban lugar al surgimiento de redes lógicas con topología de mundo pequeño. Existen otras métricas que sirven para evaluar otros aspectos de las redes. En el caso de esta Tesis se tuvieron en cuenta, además del coeficiente de clustering, la *distribución de grados* y la *descomposición en k -núcleos*.

El grado (degree) de un nodo es la cantidad de nodos vecinos que posee. El análisis de este valor permite revelar características interesantes tales como verificar si una red es libre de escala o la existencia de nodos *hubs* (también conocidos como centros de dispersión). La figura 5.5 muestra la distribución de grados (en inglés, *Degree Distribution*) en una escala logarítmica para la red física y la red lógica resultante de la primera y la décima ejecución del algoritmo 4.3.3. El análisis de esta distribución en las redes lógicas revela una característica denominada conexión preferencial (en inglés, *preferential attachment*) [BA99], lo que significa que los nodos tienden a conectarse preferencialmente

con nodos que tienen mayor grado, dando lugar al surgimiento de nodos *hubs*. Además se puede reconocer una distribución de potencias en la red lógica de la primera corrida, aunque a medida que evolucionan las ejecuciones la distribución se aleja de la distribución de potencias. Esto se debe a que a medida que aumenta el aprendizaje global de la red, más y más nodos tienden a alcanzar niveles altos de conectividad.

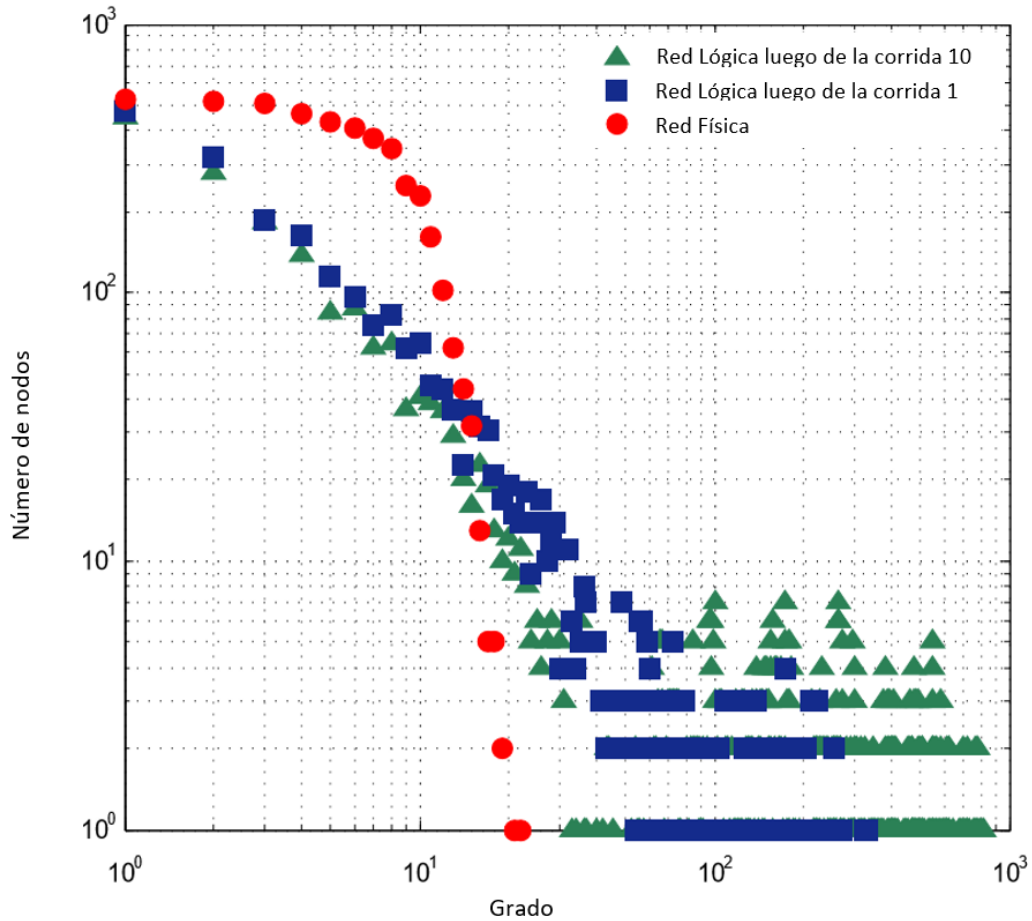


Figura 5.5: Distribución de grados para la red física y las redes lógicas resultantes de la primera y de la décima ejecución ($|N| = 5000$).

Para concluir con el estudio de las redes lógicas, se consideró la descomposición en k -núcleos [Sei83]. H_k es un k -núcleo de un grafo G si el grado de todos los nodos en H_k es mayor o igual a k y H_k es el máximo sub-grafo de G con esta propiedad. La descomposición en k -núcleos de una red permite medir la *cohesión* de la red. Si la red es *cohesiva* es posible eliminar alguno de sus participantes sin fragmentar el subconjunto. Esto además implica que los mensajes pueden alcanzar sus destinos por diferentes caminos

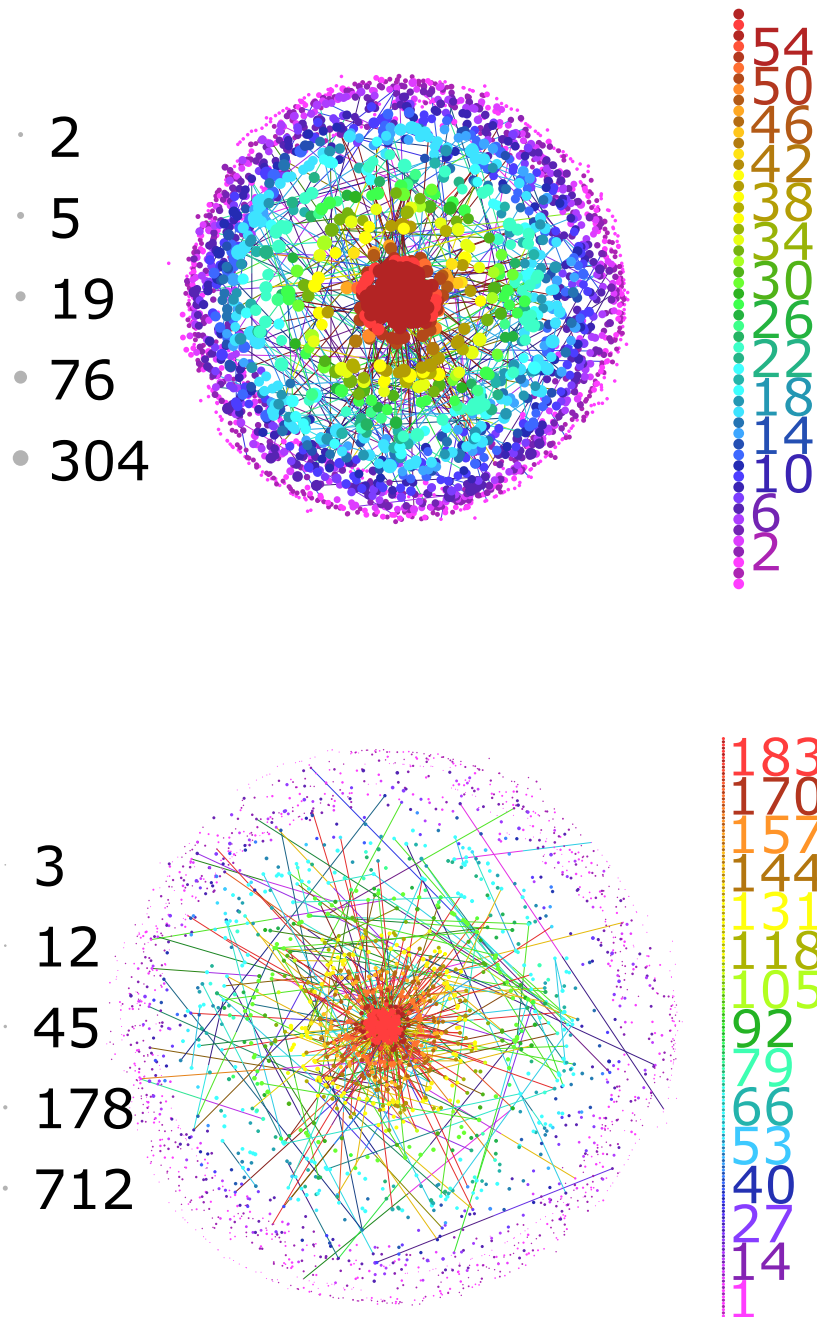


Figura 5.6: Visualización de la descomposición en k -núcleos para las redes lógicas obtenidas de la primera (arriba) y de a décima ejecución (abajo) del algoritmo 4.3.3 ($N = 5000$).

alternativos. La herramienta LaNet-vi se utilizó para obtener las visualizaciones de la descomposición en k -núcleos de la red lógica resultante de la ejecución del algoritmo 4.3.3 en una red de 5000 nodos. La figura 5.6 muestra el análisis de las redes lógicas obtenidas en la primera y la décima ejecución del algoritmo. En estas figuras el tamaño de un nodo representa su grado (leyenda izquierda) y los colores representan los núcleos (leyenda derecha). Los núcleos marcados con mayor numeración son las componentes más internas de la red. Ambas visualizaciones revelan estructuras jerárquicas y alta cohesividad, que son propiedades clave que dan lugar a un comportamiento tolerante a fallos. Sin embargo, los núcleos obtenidos durante la décima corrida tienen la propiedad de estar más densamente conectados que los obtenidos durante la primera ejecución, lo que implica que la red lógica va presentando mayores capacidades para el ruteo de los mensajes a medida que los nodos van adquiriendo más conocimiento.

5.1.5. Topología de las comunidades semánticas

Como se pudo apreciar en las secciones anteriores, los algoritmos propuestos permiten rutear consultas en una red P2P de manera eficiente gracias a determinados patrones de comunicación. Estos patrones podrían dar lugar al surgimiento de *comunidades semánticas* producto de la interacción de los nodos. La figura 5.7 muestra una red lógica compuesta por 1000 participantes obtenida como resultado de la ejecución del algoritmo 1.0. En este caso, los nodos solamente conocen a sus vecinos físicos sin tener en cuenta sus intereses. Si se tiene en cuenta que en estos gráficos, obtenidos con la herramienta JUNG, los colores similares representan tópicos similares, se puede observar que en esta red no se evidencia la existencia de comunidades semánticas. Por otro lado, las redes presentadas en las figuras 5.8 y 5.9, resultantes de las ejecuciones número 1 y 10 del algoritmo 4.3.3, indican claramente una división natural de la red en grupos de nodos con temas similares. Mientras que un alto coeficiente de clustering es un indicador de una alta inter-conectividad entre los nodos de una red, esta visualización permite reconocer que estas conexiones dan lugar a *comunidades semánticas* ya que estos núcleos de inter-conectividad están formados por nodos que comparten intereses.

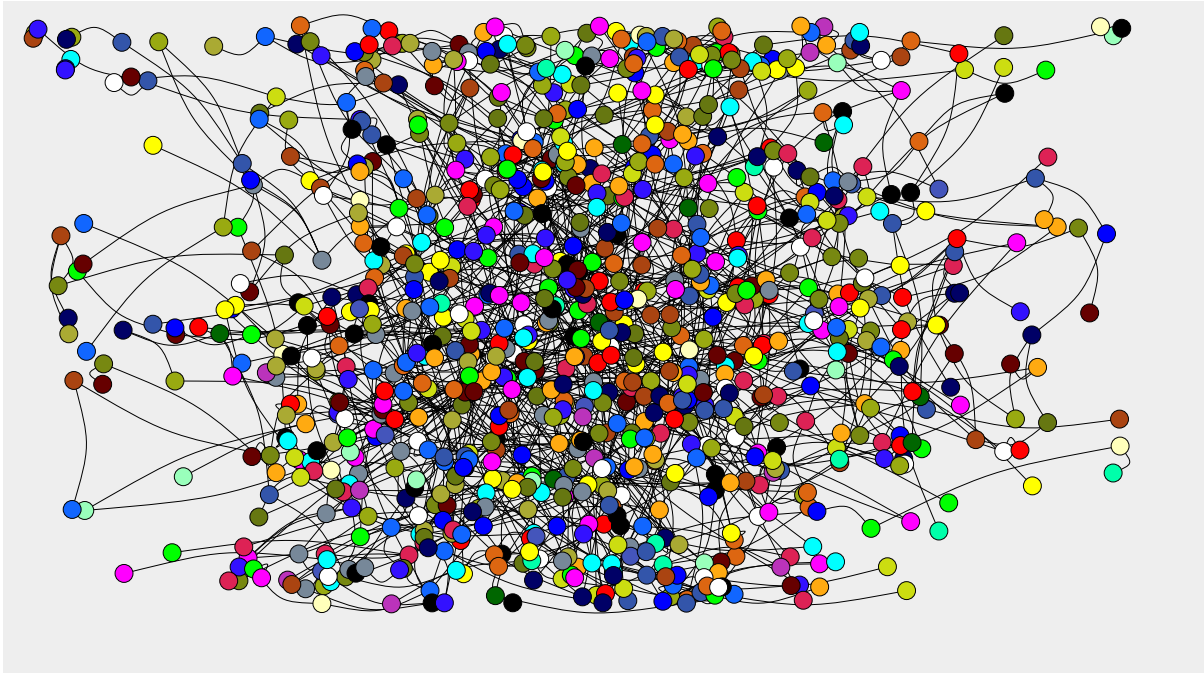


Figura 5.7: Red lógica obtenida a partir de la ejecución del algoritmo 1.0 (ejecución = 10, $|N|=1000$).

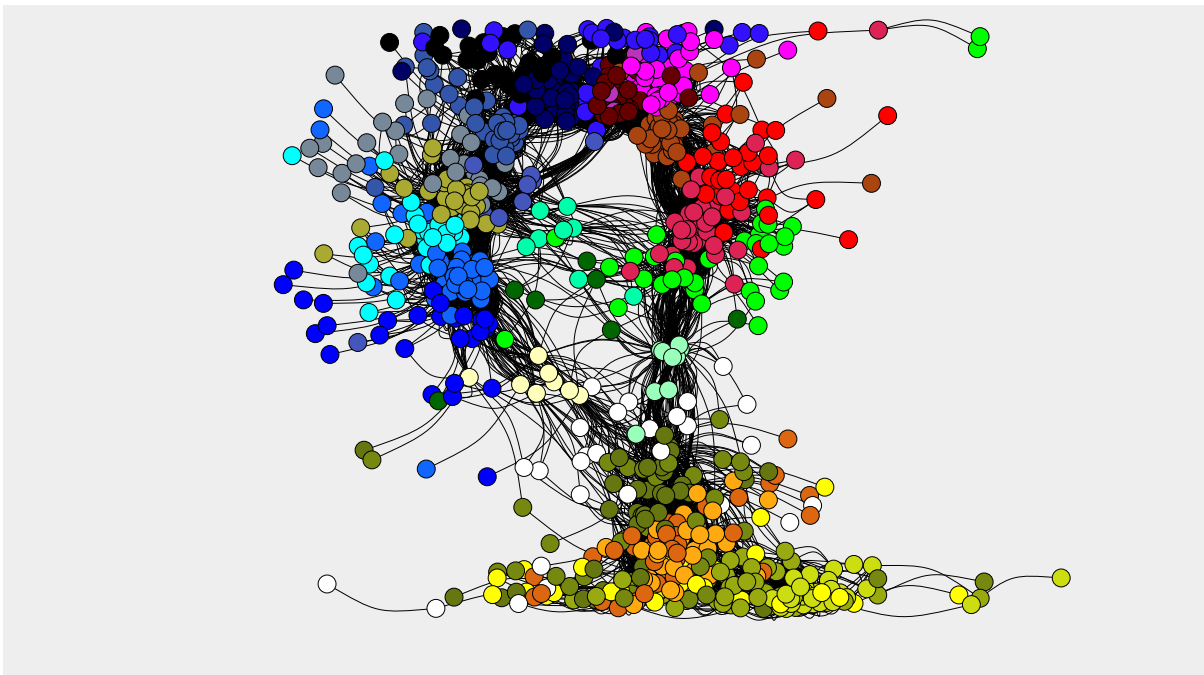


Figura 5.8: Red lógica obtenida a partir de la ejecución del algoritmo 4.3.3 (ejecución = 3, $|N|=1000$).

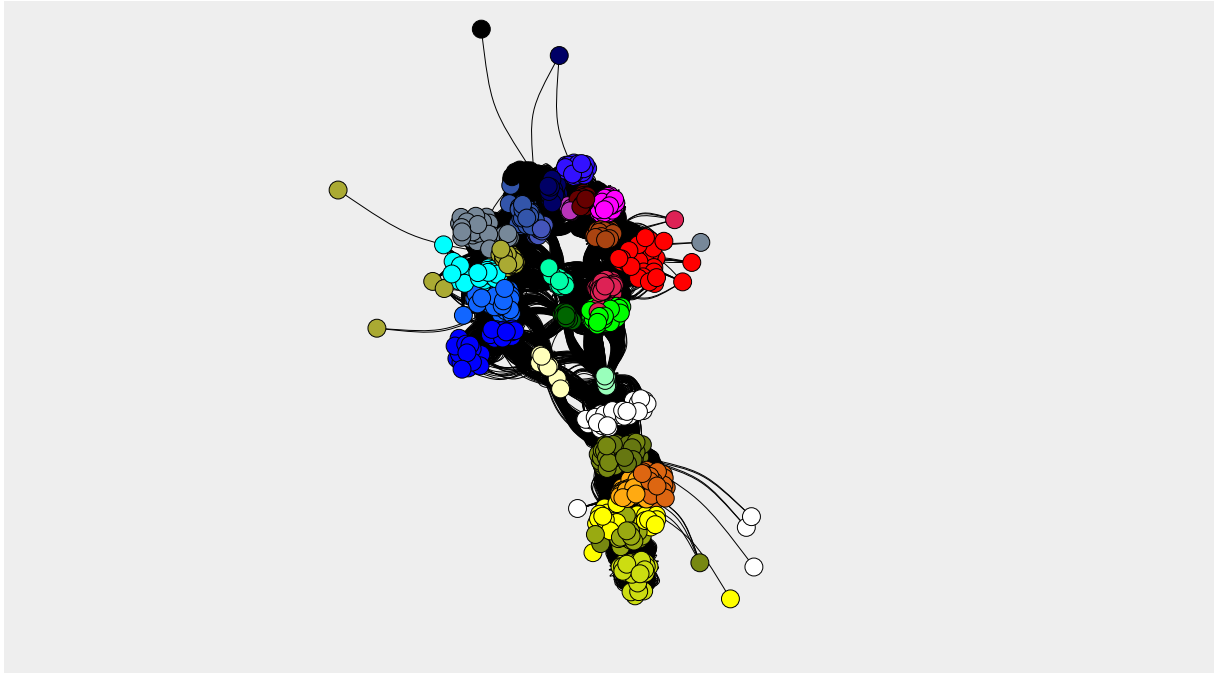


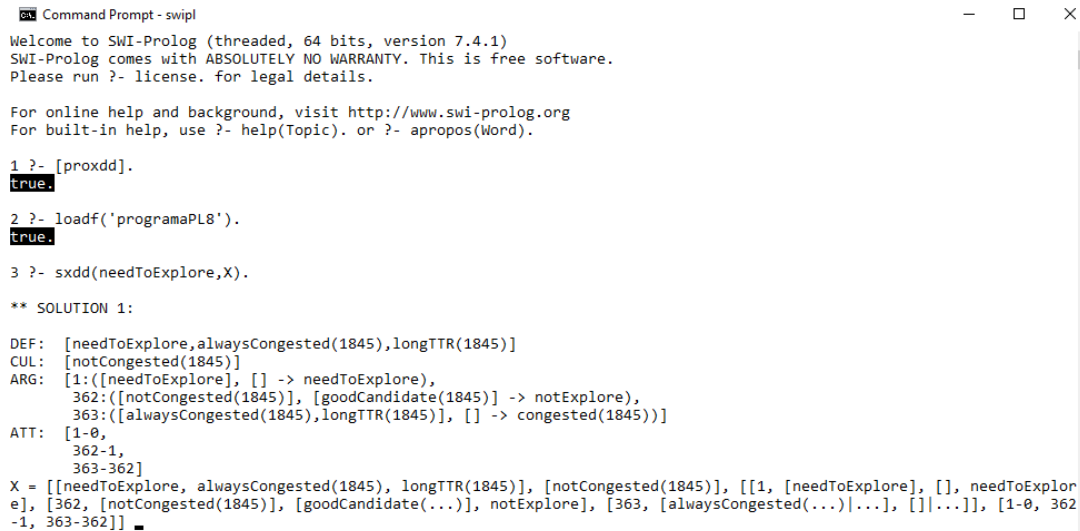
Figura 5.9: Red lógica obtenida a partir de la ejecución del algoritmo 4.3.3 (ejecución = 7, $|N|=1000$).

5.2. Evaluación del sistema Argumentativo

ArgP2P es un framework que combina dos comportamientos: un comportamiento reactivo implementado en el lenguaje C++ (algoritmo 4.3.3) y un comportamiento argumentativo implementado en Prolog.

Para programar el *Sistema Argumentativo de toma de decisiones* (ADMS) del framework presentado en el marco de esta Tesis, se tuvo en cuenta el conocimiento de cada uno de los participantes de la red. En cada nodo corre el mismo algoritmo 4.3.3 programado en C++ que conforma el sistema reactivo de toma de decisiones (RDMS). El RDMS se ejecuta cada vez que un mensaje llega a un nodo, pero el ADMS se ejecuta según una determinada frecuencia que se configura mediante el parámetro *ttarg*. Cuando este sistema es ejecutado se invoca a un programa Java que toma la tabla TN del nodo en cuestión y a partir de dicho conocimiento construye un programa ABA. Finalmente para que este programa ABA se pueda ejecutar se utiliza la herramienta *Proxdd*³ que ejecuta el algoritmo SXDD para la argumentación basada en presunciones (en inglés,

³<http://www.doc.ic.ac.uk/~rac101/proarg/proxdd.html>



```

Command Prompt - swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 7.4.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- [proxdd].
true.

2 ?- loadf('programaPL8').
true.

3 ?- sxdd(needToExplore,X).

** SOLUTION 1:

DEF: [needToExplore,alwaysCongested(1845),longTTR(1845)]
CUL: [notCongested(1845)]
ARG: [1:[needToExplore], [] -> needToExplore),
      362:[notCongested(1845)], [goodCandidate(1845)] -> notExplore),
      363:[alwaysCongested(1845),longTTR(1845)], [] -> congested(1845))]
ATT: [1-0,
      362-1,
      363-362]
X = [[needToExplore, alwaysCongested(1845), longTTR(1845)], [notCongested(1845)], [[1, [needToExplore], [], needToExplore], [362, [notCongested(1845)], [goodCandidate(...)], notExplore], [363, [alwaysCongested(...)]...], []|...]], [1-0, 362-1, 363-362]]

```

Figura 5.10: Ejemplo de funcionamiento de la herramienta *Proxdd*: encontrando argumentos a favor de explorar.



```

Command Prompt - swipl
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\anicolini>cd Desktop\proxdd\code

C:\Users\anicolini\Desktop\proxdd\code>swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 7.4.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- [proxdd].
true.

2 ?- loadf('programaPL4').
true.

3 ?- sxdd(needrToExplore,X).
false.

4 ?-

```

Figura 5.11: Ejemplo de funcionamiento de la herramienta *Proxdd*: sin argumentos a favor de explorar.

Assumption-Based Argumentation). Esta herramienta es de licencia libre, totalmente desarrollada en Prolog y con una documentación concisa que se puede obtener desde su página web. *Proxdd* se encarga de ejecutar el programa ABA generado y utilizando la consulta *sxdd(necesidadDeExplorar, X)* el programa retornará uno o más argumentos que soporten dicha decisión en caso de encontrarse garantizada o en cualquier otro caso retornará falso tal como se muestra en las figuras 5.10 y 5.11 .

Como se puede apreciar en la figura 5.12 en *ArgP2P* se ven involucrados tres lengua-

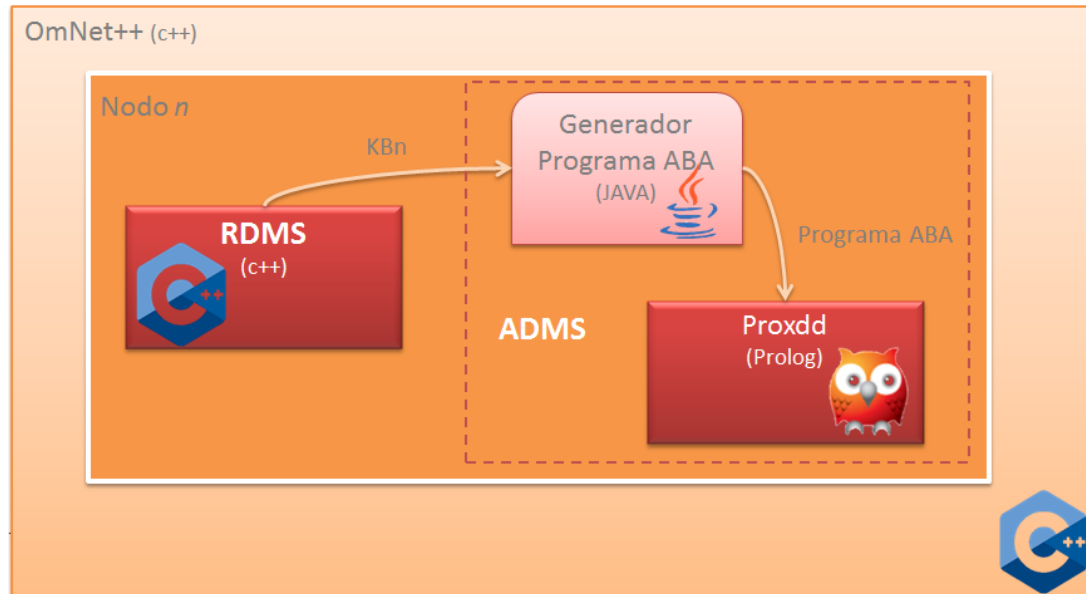


Figura 5.12: Esquema de la implementación de ArgP2P.

```

myAsm(notCongested(4342)).
myAsm(notCongested(3245)).
myAsm(notCongested(3676)).
myAsm(notCongested(4260)).
...
myAsm(goodCandidate(3646)).
myAsm(goodCandidate(4396)).
myAsm(goodCandidate(3912)).
myAsm(goodCandidate(4403)).
...
myAsm(alwaysCongested(4260)).
myAsm(alwaysCongested(3338)).
myAsm(alwaysCongested(3396)).
myAsm(alwaysCongested(4993)).
...
myAsm(longTTR(3676)).
myAsm(longTTR(3396)).
myAsm(longTTR(4993)).
myAsm(longTTR(4620)).
...
contrary(notCongested(4342), congested(4342)).
contrary(notCongested(3245), congested(3245)).
contrary(notCongested(3676), congested(3676)).
contrary(notCongested(4260), congested(4260)).

contrary(goodCandidate(3646), badCandidate(3646)).
contrary(goodCandidate(4396), badCandidate(4396)).
contrary(goodCandidate(3912), badCandidate(3912)).
contrary(goodCandidate(4403), badCandidate(4403)).
...
contrary(alwaysCongested(3245), neverCongested(3245)).
contrary(alwaysCongested(3249), neverCongested(3249)).
contrary(alwaysCongested(3676), neverCongested(3676)).
contrary(alwaysCongested(4260), neverCongested(4260)).
...
contrary(longTTR(3676), lowTTR(3676)).
contrary(longTTR(3396), lowTTR(3396)).
contrary(longTTR(4993), lowTTR(4993)).
contrary(longTTR(4620), lowTTR(4620)).
...
myRule(cienciasSocialesAplicadas(4342), []).
...
myRule(goodRR(4342), []).
myRule(goodRR(3396), []).
myRule(goodRR(4993), []).
...
myRule(congested(4342), [longTTR(4342), alwaysCongested(4342)]).
...
myRule(reliable(4342), [goodRR(4342), goodCandidate(4342)]).
...
myRule(notExplore, [forward(4342)]).

```

Figura 5.13: Fragmentos de código de un programa ABA generado por un nodo en *ArgP2P*.

jes de programación: C++ para el RDMS y Java y Prolog para el ADMS. Si bien se ha intentado optimizar la conexión entre los lenguajes para que sea altamente eficiente, el establecimiento de dicha conexión es un proceso que consume una considerable cantidad de tiempo. Es por ello que el parámetro *ttarg*, que permite establecer qué tan seguido se debe ejecutar el ADMS, es tan importante. Cuando se invoca al ADMS el framework realiza la conexión entre estos tres lenguajes mediante sockets, lo que lo hace bajar su rendimiento respecto al tiempo de ejecución. En los momentos en el que ADMS no se ejecuta el sistema sólo corre en C++ lo que lo hace altamente eficiente dadas las características propias de este lenguaje. En la figura 5.13 se muestra un programa ABA generado por el módulo Java que posteriormente será ejecutado por Proxddd.

Evaluar el ADMS por sí solo no es una cuestión trivial. Para realizar dicha evaluación se tomó una tabla TN de un nodo que presentaba características que dan indicio al *Problema de las Comunidades Cerradas* (cuando ciertos nodos sólo conocen a los nodos de la comunidad a la que pertenecen). Tomando esta tabla como entrada para el programa Java, se obtuvo el programa ABA correspondiente. Un análisis visual del mismo permitió corroborar que cada una de las componentes de este programa se generó con la sintaxis y semántica adecuada. Luego este programa es la entrada de *Proxddd* que se encarga de ejecutarlo y obtener una respuesta positiva o negativa a la necesidad de explorar. En caso de que la decisión *need_to_explore* se encuentre garantizada se verificó que los argumentos que la soportan sean válidos desde el punto de vista semántico. Dado que todas las tablas TN tienen la misma estructura no es necesario probarlo para todas, por lo tanto se realizó una verificación manual con sólo 25 de ellas (de 1000) seleccionada al azar.

5.3. Evaluación integral

A lo largo de este capítulo se han presentado los resultados obtenidos mediante las simulaciones de los algoritmos inteligentes y la evaluación del sistema argumentativo cuyo objetivo es minimizar el impacto del *Problema de las Comunidades Cerradas*. *ArgP2P* es un framework que está compuesto por dos componentes: una componente reactiva (RDMS) conformada por el algoritmo 4.3.3 y otra argumentativa (ADMS) conformada por un programa ABA. Luego de evaluar el correcto funcionamiento de ambas componentes por separado se integró todo en un gran sistema que da lugar a lo que se denominó *ArgP2P*. En esta sección se presentará una evaluación de *ArgP2P* con sus dos sistemas funcionando

de manera simultánea. El framework en cuestión se encuentra en su mayoría implementado en el lenguaje C++ lo que hace muy fácil su integración con el simulador OMNeT++, pero a la hora de implementar el ADMS se utilizaron dos sub-programas más, uno escrito en Java y otro en Prolog (tal como se explicó en la sub-sección anterior). Esta interacción entre mensajes agrega una complejidad extra al sistema. Es por ello que es muy importante la evaluación de los resultados obtenidos. El propósito de *ArgP2P* es mejorar el rendimiento de las comunicaciones en un sistema P2P, por lo que es sumamente importante verificar que su complejo funcionamiento no afecte de manera significativa el tiempo de respuesta. Para poder ajustar qué tan seguido se ejecuta el ADMS, que es el que involucra la comunicación entre los tres lenguajes, se utiliza el parámetro *ttarg* introducido en el capítulo anterior.

Para realizar esta evaluación integral de *ArgP2P* se realizaron distintas simulaciones usando datos realistas para comparar el rendimiento de los algoritmos propuestos. Las simulaciones se llevaron a cabo sobre una red física de 1000 nodos. Siguiendo la misma metodología que en las simulaciones previas, esta red fue generada usando el modelo Erdős-Rényi [ER59]. El parámetro p de este modelo fue establecido en forma tal de que cada nodo tenga un grado en promedio de 10 (ej., para $|N| = 1000$, $p = 0.002$).

Las simulaciones llevadas a cabo permitieron analizar el impacto de la componente argumentativa en el proceso de búsqueda temática. En total se realizaron cuatro simulaciones: “Simulación 1” corresponde a diez ejecuciones de *ArgP2P* sin utilizar argumentación en absoluto; “Simulación 2” corresponde a diez ejecuciones de *ArgP2P* con el parámetro *ttarg* establecido con un valor pequeño (50); “Simulación 3” corresponde a diez ejecuciones de *ArgP2P* con el parámetro *ttarg* establecido con un valor intermedio (100) y por último la “Simulación 4” corresponde a diez ejecuciones de *ArgP2P* con el parámetro *ttarg* establecido con un valor alto (250).

El principal objetivo de esta comparación es determinar cuándo y bajo qué circunstancias es apropiado el uso de la componente argumentativa de *ArgP2P*. Cuando el sistema se comporta adecuadamente se deberían visualizar redes lógicas cuyas comunidades exhiban el correcto uso de los patrones de comunicación propuestos. La tabla 5.3 muestra la cantidad de veces que la componente ADMS de *ArgP2P* fue ejecutada y respondió afirmativamente en cada corrida de cada simulación.

La tabla 5.3 junto con la figura 5.14 muestran el coeficiente de clustering promedio obtenido mediante las redes lógicas resultantes de cada corrida. Los resultados presenta-

Exploraciones por ejecución										
	Ejecución									
Sim	1	2	3	4	5	6	7	8	9	10
1	-	-	-	-	-	-	-	-	-	-
2	0	212	0	0	0	0	0	17	0	0
3	283	0	0	0	625	0	0	0	0	0
4	0	325	128	45	411	256	589	211	126	89

Tabla 5.8: Comparativa del número de veces en la que cada simulación exploró la red.

Coeficiente de Clustering										
	Ejecución									
Sim	1	2	3	4	5	6	7	8	9	10
1	0.25	0.48008	0.54531	0.60104	0.71743	0.72475	0.73261	0.73835	0.74069	0.74069
2	0.2269	0.50485	0.54002	0.60102	0.701778	0.71998	0.73861	0.74605	0.74896	0.74996
3	0.23698	0.481236	0.57322	0.61334	0.72857	0.72871	0.74073	0.7406	0.75104	0.75106
4	0.32005	0.56321	0.60254	0.69541	0.72056	0.73986	0.75001	0.75625	0.76985	0.76989

Tabla 5.9: Comparativa del coeficiente de clustering promedio en cada ejecución de cada simulación.

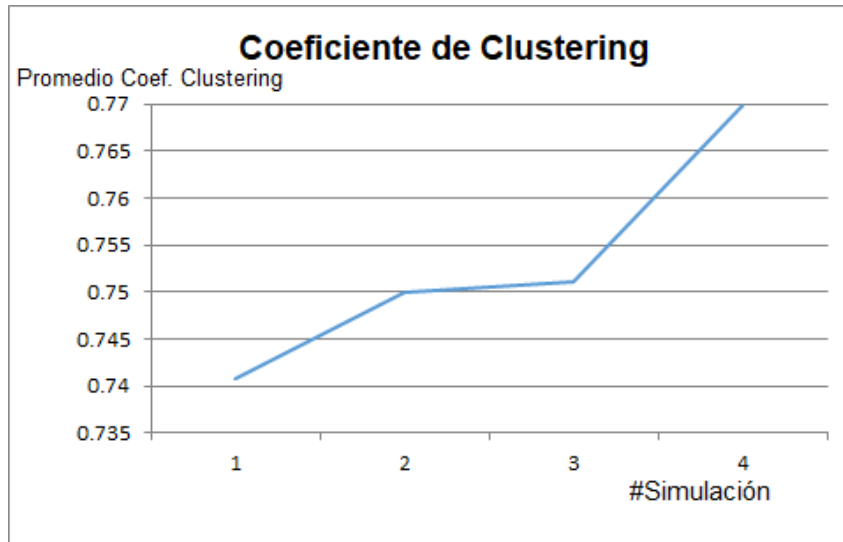


Figura 5.14: Comparativa del coeficiente de clustering obtenido en la última ejecución de cada una de las simulaciones.

dos en la tabla en cuestión indican que el coeficiente de clustering aumenta en aquellas simulaciones donde más exploraciones se han realizado. Esto se produce porque al explorar, cada nodo amplía su conocimiento respecto al resto de la red dando lugar al aumento en el conocimiento global de la red que se ve reflejado en el coeficiente de clustering.

Teniendo en cuenta que en los sistemas P2P el tiempo de respuesta se ve severamente afectado por la cantidad de tiempo que tarda la consulta en llegar al nodo que le puede dar respuesta, se consideró en este análisis la cantidad de saltos en promedio para encontrar una respuesta. La tabla 5.10 muestra la cantidad de saltos en promedio de las 10 ejecuciones de cada una de las simulaciones. Los resultados obtenidos son relativamente

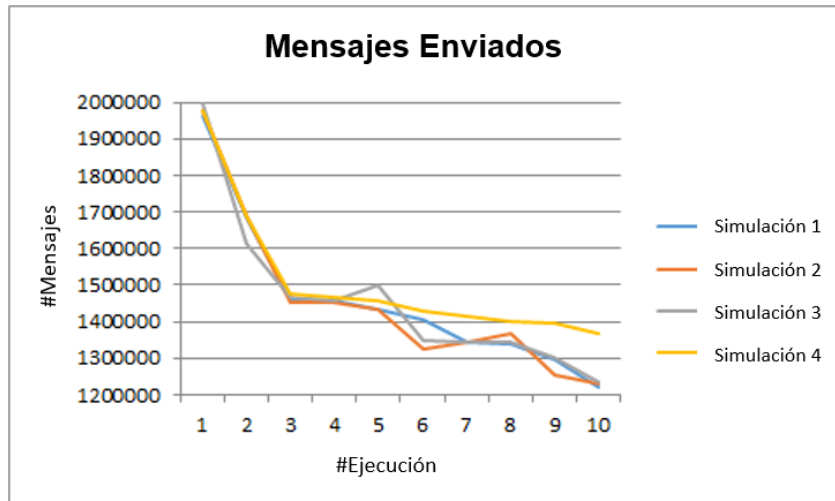


Figura 5.15: Comparativa de la cantidad de mensajes enviados.

similares pero se puede observar un pequeño incremento en aquellas simulaciones que tuvieron mayor cantidad de exploraciones. Claramente esto se debe a que al realizarse una exploración, el mensaje se propaga hacia los vecinos sin tener en cuenta cuáles pueden ser los nodos potencialmente útiles.

Simulación	Salto
Sim 1	1,998
Sim 2	2,012
Sim 3	2,3
Sim 4	2,225

Tabla 5.10: Cantidad de saltos en promedio para encontrar una respuesta.

En cualquier sistema de búsqueda el usuario requiere información en el menor tiempo posible. Existen distintos factores que determinan el tiempo de respuesta, entre ellos la congestión. Una red congestionada tardará más en dar respuesta a una determinada consulta que otra que no lo está. Es por ello que reducir la cantidad de mensajes que circulan por la red es un requerimiento sumamente importante en este tipo de sistemas. Por este motivo, se consideró en esta evaluación la cantidad total de mensajes involucrados en las simulaciones. La figura 5.15 junto con la tabla 5.3 muestran estos resultados. Se hace visible en ellos que la cantidad de mensajes es mayor en aquellas simulaciones donde se realizaron exploraciones, pero el incremento no es un valor realmente significativo

Mensajes enviados										
	Ejecución									
Sim	1	2	3	4	5	6	7	8	9	10
1	1964130	1685955	1462712	1458515	1432031	1405364	1345714	1340069	1296254	1221345
2	1975284	1686548	1453657	1452687	1435510	1326637	1342003	1368362	1252031	1229562
3	1995284	1611641	1467850	1455661	1499586	1349621	1344231	1345001	1301208	1234008
4	1975325	1687510	1475002	1465163	1456988	1428025	1416865	1400889	1396885	1368428

Tabla 5.11: Comparativa de la cantidad de mensajes enviados.

Consultas satisfechas										
	Ejecución									
Sim	1	2	3	4	5	6	7	8	9	10
1	128	97	92	95	91	89	83	87	95	91
2	108	121	86	98	99	92	99	105	101	89
3	147	102	95	93	147	86	82	85	97	93
4	135	123	98	125	126	138	112	116	122	136

Tabla 5.12: Número total de consultas satisfechas en cada ejecución.

Los sistemas P2P tienen distintas características, y más allá del algoritmo de ruteo que se utilice, el principal objetivo del usuario es encontrar una respuesta a su consulta. Es por esto que se tuvo en cuenta el número de consultas satisfechas que se muestran en la tabla 5.3. Por cada corrida de cada simulación se efectuaron 150 consultas, de las cuales la mayor cantidad fue satisfecha en aquellas corridas donde se realizaron exploraciones. Los algoritmos que inundan la red de consultas por lo general tienen un alto grado de consultas satisfechas. Como contrapartida de esto, el hecho de inundar la red tiene un impacto negativo respecto al rendimiento general del sistema. Como resultado del aumento de la congestión, distintos aspectos se ven afectados siendo la cantidad de saltos para encontrar una respuesta uno de ellos. Particularmente nuestras simulaciones demostraron que la mayoría de las consultas fueron satisfechas en aquellas corridas donde se realizaron exploraciones pero sin penalizar en forma significativa la cantidad de saltos en promedio.

En la figura 5.16 se presenta una comparación visual de las redes lógicas resultantes en la décima corrida de las simulaciones 1 y 4 respectivamente. Gráficamente el *Problema de las Comunidades Cerradas* se evidencia cuando nodos de la misma temática (color) se encuentran desconectados o formando comunidades totalmente independientes. Este fenómeno se puede observar en los nodos azules en la red de la parte A de la figura. En la parte B de la figura se puede ver que este problema se encuentra mayormente resuelto.

La evaluación integral de este framework ha demostrado que el uso de la argumentación da lugar a una mejora respecto a la conformación de las comunidades semánticas, resultando en patrones de comunicación más efectivos. Es importante resaltar que los principales aspectos de los sistemas P2P no sólo no se ven afectados sino que algunos de ellos se ven beneficiados como es el caso de la cantidad de consultas satisfechas, que mejora notablemente en aquellas simulaciones en las que se realizaron exploraciones.

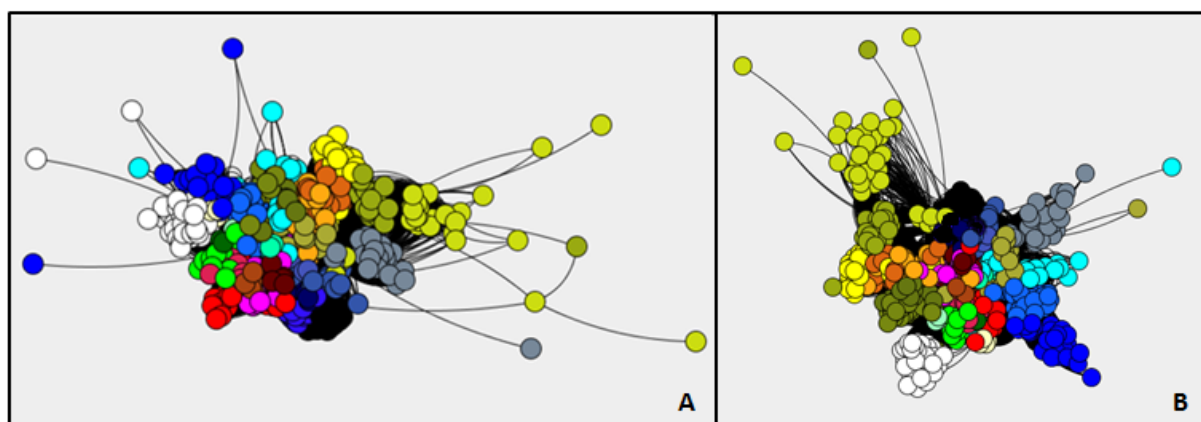


Figura 5.16: (A) Red lógica obtenida en la décima corrida de la Simulación 1. (B) Red lógica obtenida en la décima corrida de la Simulación 4.

5.4. Resumen del capítulo

A lo largo de este capítulo se han presentado los resultados obtenidos junto con su correspondiente análisis. En primera instancia se evaluaron cada uno de los algoritmos inteligentes en forma individual, resultados que mostraron que los buenos patrones de comunicación emergieron mediante el aprendizaje de los pares respecto a los intereses de los otros participantes del sistema P2P. Además, los resultados de las simulaciones corridas en distintas escalas han mostrado que los algoritmos que exhiben mejor comportamiento son aquellos que presentan mayor colaboración entre pares, lo que significa que cuando un nodo incorpora conocimiento relevante lo comparte con el resto de su comunidad. Los algoritmos 4.3.3 y 5.0 son los que implementan en mayor medida este comportamiento. A partir de los resultados de sus simulaciones se pudo observar que si bien en la primera ejecución el número de mensajes enviados es alto, luego decae a medida que las simulaciones avanzan y el aprendizaje se incrementa. Para avanzar con la evaluación de los resultados obtenidos, se llevó a cabo un exhaustivo análisis de las redes lógicas resultantes de las simulaciones. En todos los algoritmos (excepto el 1.0 que carece de un sistema inteligente) se obtuvieron redes lógicas en las que se evidencia el surgimiento de *comunidades semánticas* producto de la comunicación basada en intereses que tienen los participantes del sistema. Las mismas se pueden observar a simple vista (mediante los gráficos obtenidos mediante la herramienta JUNG) y al mismo tiempo su alto coeficiente de clustering lo demuestra en forma cuantitativa. Al mismo tiempo los resultados muestran que estas redes presentan un diámetro pequeño (aproximado por la cantidad de saltos para encontrar una respuesta). Esta característica junto con el alto coeficiente de clustering son claras evidencias de que las redes lógicas resultantes de las simulaciones de los algoritmos inteligentes presentan una *topología de mundo pequeño*. Otro resultado sumamente importante producto de las simulaciones es que los algoritmos propuestos producen un impacto similar en su comportamiento independientemente de que el tamaño de la red sea de 1000 o 5000 nodos, sugiriendo de esta manera que los algoritmos escalan bien hacia redes de mayor tamaño.

El llamado *Problema de las Comunidades Cerradas* no ha pasado desapercibido a lo largo de esta evaluación. En los gráficos de las redes lógicas se pueden visualizar comunidades y nodos desconectados, que como tienen intereses en común (ilustrado mediante el mismo color o colores similares en el gráfico) deberían estar en la misma comunidad. Como se propuso en el Capítulo 4, el framework *ArgP2P* tiene la capacidad de mitigar los efectos de este problema. Las simulaciones de *ArgP2P* han demostrado que el uso mo-

derado de la argumentación, dada por un programa ABA, soluciona casi en su totalidad el *Problema de las Comunidades Cerradas* sin penalizar en forma significativa aspectos importantes tales como la cantidad de saltos en promedio para encontrar una respuesta y la cantidad de mensajes que circulan en la red.

Capítulo 6

Conclusiones y trabajo a futuro

En este capítulo se presentan las principales conclusiones extraídas durante el desarrollo de esta Tesis junto con una descripción de las problemáticas y líneas investigativas sobre las que se proyecta seguir trabajando.

6.1. Conclusiones

Las tecnologías distribuidas P2P ofrecen una posible solución al problema de la centralización de la información por parte de grandes corporaciones. Estas tecnologías favorecen el control sobre la información, permitiendo que los servidores estén bajo el control de los propios usuarios. De esa manera, los servidores distribuidos de un sistema P2P puro se comunican directamente entre sí, sin la necesidad de recurrir a un servidor central. Esto permite que los datos y el control sobre los mismos se encuentren distribuidos entre todos los usuarios y sus servidores. Para que este tipo de servicios funcione adecuadamente se debe garantizar un tiempo de respuesta razonable y una alta posibilidad de encontrar una respuesta a cualquier consulta iniciada por un usuario. Es por este motivo que se han desarrollado distintos tipos de algoritmos para el ruteo de consultas en redes P2P. En sus orígenes, Internet fue diseñado basada en los principios de la cooperación y de la buena ingeniería, y se manejaba como una red P2P pura. En este escenario descentralizado era indispensable contar con algoritmos para realizar búsquedas. A medida que la estructura de Internet se volvió más rígida, surgieron los algoritmos para búsquedas en redes estructuradas y semi-estructuradas, donde la colaboración entre los miembros de la red no es

un aspecto importante. En los últimos tiempos, las redes P2P puras han ganado importancia nuevamente, desencadenando en nuevas investigaciones respecto a algoritmos que incorporan aspectos semánticos derivados de los perfiles e intereses de los usuarios. Estos aspectos semánticos, pueden ser convenientemente explotados para mejorar el rendimiento de los algoritmos de ruteo, minimizando el tráfico en la red y mejorando el tiempo que se tarda en obtener una respuesta para una determinada consulta.

La clasificación de redes P2P en estructuradas, semi-estructuradas y no estructuradas resulta en que los algoritmos de búsqueda que se utilicen para cada tipo de red posean distintas características, pero manteniendo entre sí aspectos comunes. Es por eso que en el marco de esta Tesis se presentó un extenso trabajo de revisión de literatura junto con un análisis comparativo respecto a los aspectos más relevantes de los principales sistemas desarrollados hasta el momento.

Distintos fundamentos teóricos y matemáticos se ven involucrados en lo que respecta a búsqueda en redes P2P. El concepto de red P2P es fundamental en el marco de esta Tesis, motivo por el cual se introdujo una definición apropiada junto con una novedosa notación que permite capturar los aspectos dinámicos de este tipo de sistemas. Distintas métricas se ven involucradas a la hora de evaluar este tipo de sistemas, como por ejemplo el coeficiente de clustering, que permite analizar en forma cuantitativa la formación de comunidades semánticas a partir de las redes lógicas obtenidas mediante la interacción entre los pares de la red.

Como contribución principal se presentó una serie de algoritmos que explotan el hecho de que si se dirigen las búsquedas según los intereses de los participantes, emergerán patrones de comunicación eficientes que llevan directamente a la mejora en el rendimiento del sistema. Cada uno de estos algoritmos fue presentado junto a un ejemplo ilustrativo de su comportamiento ante situaciones particulares. Además se presentó una tabla en la que se comparan los aspectos en común entre todos ellos. Una de las limitaciones observadas es lo que se describió como el *Problema de las Comunidades Cerradas*. Bajo este escenario uno o más nodos pueden quedar separados de su comunidad, o hasta incluso puede existir una misma comunidad dividida en dos o más comunidades pequeñas. La emergencia de este tipo de comunidades desconectadas es perjudicial para la interacción de los participantes bajo un sistema de búsqueda temática. Por lo tanto, es necesario proponer algún mecanismo para atenuar los efectos de este problema. El framework *ArgP2P* ayuda a minimizar el impacto del *Problema de las Comunidades Cerradas*, con un no-

vedoso modelo de búsqueda temática en redes P2P donde cada nodo de la red tiene la habilidad de combinar un comportamiento reactivo con un comportamiento racional (argumentativo). La máquina de inferencia argumentativa se provee mediante un programa lógico que posibilita alcanzar decisiones más acertadas respecto a qué pasos seguir tras la llegada de un mensaje de consulta a un nodo. Este framework brinda un parámetro ajustable denominado *ttarg* que permite hacer a los participantes más “reactivos” o más “racionales” dependiendo de qué tan frecuentemente se ejecute el sistema argumentativo.

ArgP2P, la propuesta principal de esta Tesis, es un poderoso framework que busca garantizar las principales propiedades de los sistemas P2P minimizando la cantidad de saltos para encontrar una respuesta, dando de este modo un tiempo de respuesta tolerable. Para lograr esto, la búsqueda está guiada por temáticas y un mecanismo de aprendizaje hace que la red en su totalidad incremente su conocimiento a medida que aumentan las interacciones entre sus participantes. *ArgP2P* está compuesto por dos sistemas: el sistema reactivo de toma de decisiones (RDMS) y el sistema argumentativo de toma de decisiones (ADMS). Ambos sistemas fueron evaluados individualmente para asegurar su correcto funcionamiento. Los resultados han mostrado que aquellos algoritmos en los que un nodo comparte su conocimiento con su comunidad presentan mejores prestaciones. Las evaluaciones realizadas demuestran que el ADMS atenúa el *Problema de las Comunidades Cerradas*, lo que se vió reflejado en la evaluación integral con las redes lógicas resultantes de la ejecución de *ArgP2P*. Los resultados han demostrado que el uso excesivo del sistema argumentativo eleva el tiempo de respuesta ya que circulan más mensajes en la red, lo que no es un aspecto deseable en una red P2P. De todos modos también se ha detectado que el uso moderado del sistema argumentativo no provoca un aumento significativo en la cantidad de mensajes y la penalización en el tiempo de respuesta es mínimo. En general, el uso moderado de la argumentación muestra mejoras respecto al *Problema de las Comunidades Cerradas* sin penalizar significativamente ninguno de los otros aspectos, logrando de este modo una mejor experiencia para los usuarios del sistema.

Una cuestión a resaltar de este framework es que es lo suficientemente general como para cambiar alguna de sus dos componentes para evaluar otro tipo de sistemas, ya sean reactivos o argumentativos. El sistema argumentativo de toma de decisiones fue desarrollado utilizando el framework argumentativo Assumption-Based Argumentation. Esa elección fue resultado del análisis de las facilidades que ofrece este estilo computacional originado en la programación lógica y que tiene numerosas ventajas sobre otros mecanis-

mos computacionales. Las ventajas se deben principalmente al fino nivel de granularidad proporcionada por la forma en la que se construyen los argumentos y en la que se determina su aceptabilidad. Por varios motivos, incluido el mencionado anteriormente, ABA se ha vuelto particularmente aplicable en el contexto de problemas reales. ABA posee distintas semánticas, particularmente en el marco de esta Tesis se ha utilizado la semántica escéptica.

La obtención de resultados y el análisis de los mismos ha sido fundamental para comprobar el rendimiento de *ArgP2P*. En primera instancia se evaluaron cada uno de los algoritmos inteligentes en forma individual, resultados que mostraron que los buenos patrones de comunicación emergieron mediante el aprendizaje de los pares respecto a los intereses de los otros participantes del sistema P2P. Además, los resultados de las simulaciones corridas en distintas escalas han mostrado que los algoritmos que exhiben mejor comportamiento son aquellos que presentan mayor colaboración entre pares, lo que significa que cuando un nodo incorpora conocimiento relevante lo comparte con el resto de su comunidad. Los algoritmos 4.3.3 y 5.0 son los que implementan en mayor medida este comportamiento, y a partir de los resultados de sus simulaciones se pudo observar que si bien en la primera ejecución el número de mensajes enviados es alto, luego decae a medida que las simulaciones avanzan y el aprendizaje se incrementa. Para continuar con la evaluación de los resultados obtenidos, se llevó a cabo un exhaustivo análisis de las redes lógicas resultantes de las simulaciones. En todos los algoritmos (excepto el 1.0 que carece de un sistema inteligente) se obtuvieron redes lógicas en las que se evidencia el surgimiento de *comunidades semánticas* producto de la comunicación basada en intereses que tienen los participantes del sistema. Las mismas se pueden observar a simple vista (mediante los gráficos obtenidos mediante la herramienta JUNG) y al mismo tiempo su alto coeficiente de clustering lo demuestra en forma cuantitativa. Además, los resultados muestran que estas redes presentan un diámetro pequeño (aproximado por la cantidad de saltos para encontrar una respuesta). Esta característica junto con el alto coeficiente de clustering son claras evidencias de que la redes lógicas resultantes de las simulaciones de los algoritmos inteligentes presentan una *topología de mundo pequeño*. Otro resultado sumamente importante producto de las simulaciones es que los algoritmos propuestos producen un impacto similar en su comportamiento independientemente de que el tamaño de la red sea de 1000 o 5000 nodos, sugiriendo de esta manera que los algoritmos escalan bien hacia redes de mayor tamaño.

El llamado *Problema de las Comunidades Cerradas* no ha pasado desapercibido a lo largo de esta evaluación. En los gráficos de las redes lógicas se puede visualizar comunidades y nodos desconectados, que como tienen intereses en común deberían estar en la misma comunidad. Como se expuso en el Capítulo 4, el framework *ArgP2P* logra mitigar los efectos de este problema.

Durante el transcurso de desarrollo de esta Tesis, se publicaron tres artículos científicos, el primero presenta una versión preliminar de los algoritmos inteligentes [NLMC13], luego se publicó el marco teórico de *ArgP2P* [NMC15] y finalmente se publicó un análisis exhaustivo de los algoritmos inteligentes [NLMC17]. Otros dos artículos cuyo contenido es una evaluación intensiva de *ArgP2P* y una revisión de literatura, se encuentran actualmente bajo revisión.

6.2. Trabajo a futuro

Durante el desarrollo de esta Tesis se detectaron distintas problemáticas que es oportuno acatar como trabajo a futuro. Muchas de estas problemáticas ya han sido estudiadas en el contexto de redes P2P pero no específicamente en combinación con el concepto de búsqueda temática distribuida. En esta sección se describen distintos aspectos de las problemáticas detectadas y posibles maneras de dar solución (total o parcial) a las mismas.

La eficiencia en una red P2P depende de la colaboración y la interacción existente entre los pares. Esta colaboración suele estar motivada por distintos mecanismos de incentivos [HB15]. Un simple mecanismo de incentivo consiste en tomar en cuenta la contribución de un par en el proceso de búsqueda en general y determinar si este tiene un papel destacado dentro de los resultados de las búsquedas. Este enfoque normalmente funciona tanto para las búsquedas web como en redes P2P, ya que las empresas y las personas, como proveedores de información, tienen un incentivo para aparecer en lo alto de la clasificación de resultados de búsqueda [CH09]. Como parte de nuestras investigaciones futuras planeamos aplicar *ArgP2P* en el desarrollo de mecanismos de incentivos.

Las comunidades conectadas mediante tecnologías de la información representan un mecanismo poderoso a la hora de coordinar esfuerzos para capturar, refinar y compartir conocimiento y para la toma de decisiones. Si bien la web social provee una plataforma natural para avanzar en estas áreas, la misma se ve limitada en gran medida por la

centralización en el uso de la información y potencial pérdida de control sobre dicha información por parte de los usuarios, tal como es el caso de las redes sociales centralizadas Twitter y Facebook (entre otras). Dichas redes sociales crean la ilusión de que los usuarios se encuentran directamente conectados entre sí. Sin embargo, son servidores centrales pertenecientes a las compañías los que controlan todos los datos y las interacciones entre los usuarios. El problema de la falta de control de los usuarios sobre sus datos se ve agravado por los términos de servicio poco claros o con desventajas notables y por los cambios ocasionales en la política de privacidad. Las tecnologías distribuidas, entre ellas las tecnologías P2P, ofrecen una posible solución al problema de la centralización ya que permiten que los datos y el control sobre los mismos se encuentre distribuido entre todos los usuarios y sus servidores. Es interesante pensar a *ArgP2P* como una potencial red social distribuida, donde cada uno de los participantes sea un usuario con intereses propios que tenga total acceso a su información, pero al mismo tiempo sea capaz de enviar sugerencias a potenciales usuarios con intereses comunes.

Los principios colaborativos, adaptativos, transparentes y descentralizados de las tecnologías distribuidas también las vuelven apropiadas para la implementación de servicios públicos digitales, en particular en el contexto de gobierno electrónico. Este tipo de servicios necesita dotarse de una infraestructura segura, capaz de lidiar con problemas tales como la confiabilidad y el manejo de grandes volúmenes de datos. Las tecnologías distribuidas brindan una oportuna manera de enfrentar estos desafíos, siendo sus principales beneficios la descentralización en el control de la información, la escalabilidad y la alta tolerancia a fallos. Esto permite que el sistema pueda seguir funcionando aún cuando algunos de los participantes se encuentran fuera de servicio, aspecto fundamental en cualquier sistema de gobierno electrónico. Existen diversos mecanismos orientados a modelar la noción de confiabilidad o reputación en sistemas P2P [CDE08, NCR08]. En tales sistemas, el cómputo de una medida de reputación global o de medidas subjetivas de confiabilidad es una tarea difícil, que involucra mecanismos de propagación de valores a través de una red de pares. En muchos casos, esto se realiza utilizando técnicas ad-hoc que carecen de una semántica precisa. Resultados recientes [TCM⁺11, ALL⁺16] han constatado la importancia de la argumentación como herramienta versátil para manejar confiabilidad y reputación en entornos multiagentes y particularmente P2P. En tal sentido, se proyecta incorporar formalismos argumentativos dentro de nuestro modelo para capturar distintos aspectos complejos de búsqueda distribuida. Se planea abordar temas vinculados a mejorar cuestiones de confiabilidad en los sistemas de búsqueda actuales. La problemática

será atacada usando técnicas de aprendizaje semi-supervisado junto con argumentación para dotar a cada uno de los participantes de la red con un sistema de toma de decisiones inteligentes que le permita a cada uno en forma individual evaluar al resto de los pares y compartir esta evaluación con el resto de los participantes. De esta manera, los participantes menos confiables quedarían apartados del resto hasta que mejoren sus niveles de confiabilidad.

Otro ingrediente de del trabajo futuro son los sistemas de recomendación, que predicen los intereses de un usuario integrando los intereses de usuarios similares. Los primeros sistemas de recomendación tales como GroupLens [RIS⁺94] no permitían una colaboración real entre los usuarios. Los mismos no “aprendían unos de otros” ni tomaban en cuenta la noción de confiabilidad. Algunos sistemas sucesores intentaron superar algunas de estas limitaciones mediante la incorporación de mecanismos cualitativos para la generación de sugerencias confiables (e.g., [CMG09]). Más recientemente, con el advenimiento y expansión de las redes sociales en Internet, los sistemas de recomendación se han visto potenciados por la construcción de comunidades y la incorporación de mecanismos que facilitan capturar los intereses de un número masivo de usuarios (e.g. [WBS08, GS09, ECMB12]). Se proyecta seguir explotando el concepto de “localidad semántica en un sistema distribuido” como factor para predecir la eficacia en un sistema de recomendación P2P.

Apéndice A

Glosario

Búsqueda temática: Es un mecanismo de búsqueda que vincula a los datos con temáticas. De esta forma, al momento de realizar una búsqueda se tiene en cuenta el tópico de la consulta y luego la misma es dirigida hacia aquellos miembros del sistema que están vinculados con la temática en cuestión.

Broadcast: Envío de información a todos los dispositivos de una red.

Clasificador: Son algoritmos que se utilizan para etiquetar elementos en categorías conocidas. Estos algoritmos permiten ordenar o disponer en distintas clases a los elementos a partir de cierta información característica de estos.

Comunidades semánticas: Son grupos de dispositivos vinculados entre sí en forma lógica (se conocen unos a otros) que comparten intereses en común.

Congestión: La congestión en una red de datos es un ítem de calidad de servicio que se ve afectado cuando uno o más miembros de la red manipula más datos de los que puede manejar afectando así su tiempo de respuesta. Los efectos adversos de las congestiones son retrasos, pérdida de paquetes y bloqueo de nuevas conexiones.

DHT: Las tablas de hash distribuidas, conocidas por las siglas DHT (del inglés, Distributed Hash Tables), son un tipo de tablas hash que almacenan pares (clave, valor) y permiten consultar el valor asociado a una clave, en las que los datos se almacenan de forma distribuida en una serie de nodos y proveen un servicio eficiente de búsqueda que permite encontrar el valor asociado a una clave. Para esto último usa un sistema de ruteo que permite encontrar de forma eficiente el nodo en el cual está almacenada la información que se necesita.

Distribución de grados: En el estudio de grafos y redes complejas, el grado de un nodo en una red es el número de conexiones asociadas a ese nodo. Si se hiciera un recuento en una red del número de nodos por cada grado se tendría una distribución de grado que sería entendido igualmente como la distribución de probabilidad de un grado en la red. Cuando esta función de distribución de grados sigue una ley de potencias que es independiente de la escala de la red, se está en presencia de una red libre de escala.

Flooding: Es un algoritmo de búsqueda en grafos, particularmente en redes P2P. En este algoritmo, cada uno de los nodos al recibir una consulta a la que no le pueden dar respuesta la re-envía a sus propios vecinos y así sucesivamente. *Gnutella* es uno de los sistemas P2P más conocidos en su época que estaba basado en este algoritmo.

Hipercubo: Un hipercubo es un análogo n -dimensional de un cuadrado ($n = 2$) y un cubo ($n = 3$). Se trata de una figura cerrada, compacta y convexa cuyo esqueleto consiste en grupos de segmentos de línea paralelos opuestos alineados en cada una de las dimensiones del espacio, perpendiculares entre sí y de la misma longitud.

Hub: Hub o centro de dispersión es un concepto de redes de computadoras que se refiere a un nodo con un número de enlaces que excede en gran medida el promedio. La aparición de los hubs es una consecuencia de la propiedad libre de escala de las redes. Los hubs se pueden encontrar en muchas redes reales, como Brain Network o Internet.

Índice: Es una estructura que mantiene información respecto a la ubicación de contenidos. En lo que respecta a sistemas P2P descentralizados, cada nodo mantiene un índice en el que se mantiene información respecto a la ubicación de otros nodos. Los índices se clasifican en centralizados o distribuidos según donde se encuentren almacenados. En algunos sistemas, en los que el índice suele ser de gran tamaño, se distribuye el índice entre varios participantes logrando de este modo balancear la carga respecto al espacio en memoria. Cada algoritmo maneja su o sus índices según sea más conveniente: algunos tienen un sólo índice, en otros cada nodo tiene un índice propio y en otros un conjunto pequeño de nodos comparten un mismo índice.

IP: Una dirección IP es un número que identifica, de manera lógica y jerárquica, a una interfaz en red de un dispositivo que utilice el protocolo IP (en inglés, Internet Protocol), que corresponde al nivel de red del modelo TCP/IP.

Nodo: Un nodo es un miembro o participante de una red. Cada uno de estos nodos representan a dispositivos inteligentes que tengan conexión a una red, por ejemplo: servidores, computadoras personales o smartphones.

Ontología: En ciencias de la computación, una ontología es una definición formal de tipos, propiedades, y relaciones entre entidades que realmente o fundamentalmente existen para un dominio de discusión en particular. En inteligencia artificial, Web semántica, ingeniería de sistemas, ingeniería de software, bioinformática y la arquitectura de la información, entre otros, se crean ontologías para limitar la complejidad y organizar la información.

Protocolo: Un protocolo de comunicaciones es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información. Se trata de las reglas o el estándar que define la sintaxis, semántica y sincronización de la comunicación, así como también los posibles métodos de recuperación de errores.

Red libre de escala: Una red libre de escala es un tipo específico de red compleja. En una red libre de escala, algunos nodos están altamente conectados, es decir, poseen un gran número de enlaces a otros nodos, aunque el grado de conexión de casi todos los nodos es bastante bajo. Ver definición distribución de grados.

Red P2P: Una red peer-to-peer (P2P, por sus siglas en inglés) es una red de dispositivos en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás miembros de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre los dispositivos interconectados. Entre sus aplicaciones más populares se encuentran: manejo de monedas digitales, manejo descentralizado de conexiones inalámbricas (Netsukuku), intercambio de archivos y redes sociales distribuidas.

Red Social: Una red social es una estructura social compuesta por un conjunto de actores (tales como individuos u organizaciones) que están relacionados de acuerdo a algún criterio (relación profesional, amistad, parentesco, etc.). Normalmente se representan simbolizando los actores como nodos y las relaciones como líneas que los unen. El análisis de redes sociales estudia esta estructura social aplicando la teoría de grafos e identificando las entidades como “nodos” o “vértices” y las relaciones como “enlaces” o “aristas”. La estructura del grafo resultante es a menudo una red compleja. Las plataformas en Internet que facilitan la comunicación entre personas de una misma estructura social se denominan servicios de red social o redes sociales virtuales, algunos ejemplos de ellas son Facebook, Instagram y Twitter.

Ruteo: El ruteo, o encaminamiento, es la acción de buscar un camino entre todos los posibles en una red de nodos. El estudio de algoritmos de ruteo se realiza sobre redes que suelen tener distintas topologías y gran conectividad.

Sistema Híbrido: Es un concepto para modelar sistemas y procesos complejos. Una característica de la ingeniería de hoy es su creciente complejidad, que surge al vincular procesos y sistemas de diferentes índoles. En lo que respecta a redes P2P híbridas, son sistemas P2P que combinan características propias de redes P2P con distintos grados de centralización. Por ejemplo existen sistemas híbridos que trabajan con redes P2P descentralizadas sobre las que se crean capas subyacentes con cierto grado de centralización como es el caso del algoritmo *FreeNet*.

Topología: La topología de red se define como el mapa físico o lógico de una red para intercambiar datos. En otras palabras, es la forma en que está diseñada la red, sea en el plano físico o lógico. En las topologías físicas los enlaces o conexiones están dados por la forma en que los dispositivos se encuentran conectados físicamente, mientras que en las topologías lógicas los enlaces están dados por una relación de conocimiento, es decir: en una topología lógica dos dispositivos se encuentran relacionados si “uno conoce al otro”.

TTL: Tiempo de Vida o Time To Live (TTL) es un concepto usado en redes de computadores para indicar por cuántos nodos puede pasar un paquete o mensaje antes de ser descartado por la red o devuelto a su origen.

Bibliografía

- [ALL⁺16] APPLEBAUM, A., LI, Z., LEVITT, K., PARSONS, S., ROWE, J., AND SKLAR, E. I. Firewall configuration: An application of multiagent metalevel argumentation. *Argument & Computation* 7, 2-3 (2016), 201–221.

En este artículo los autores discuten cómo un framework argumentativo puede ser utilizado para analizar políticas vinculadas a los cortafuegos o *firewalls*. Se propone agregar un metanivel de razonamiento sobre la aceptabilidad de los argumentos para ayudar a razonar respecto a conflictos entre políticas de los cortafuegos.

- [ALPH01] ADAMIC, L. A., LUKOSE, R. M., PUNIYANI, A. R., AND HUBERMAN, B. A. Search in power-law networks. *Phys. Rev. E* 64 (Sep 2001), 046135.

En este artículo se presentan distintas estrategias de búsqueda local en redes cuya distribución de los grados de sus nodos respeta una distribución de potencias. Estas estrategias explotan el hecho de que este tipo de redes presenta nodos con alta conectividad o hubs, que juegan un papel central en las comunicaciones.

- [ATS04] ANDROUTSELLIS-THEOTOKIS, S., AND SPINELLIS, D. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys (CSUR)* 36, 4 (2004), 335–371.

En este artículo se presenta una revisión de literatura respecto al manejo de distintas características en sistemas P2P. Entre estas características se encuentran la seguridad, la escalabilidad, el rendimiento y el manejo de recursos. Los autores analizan la forma en la que estas características se ven reflejadas (y afectadas) por el diseño de la arquitectura P2P adoptada. También presentan compara-

ciones entre sistemas con distintos métodos de localización de contenido y de acceso a la información.

- [AWMM06] AKAVIPAT, R., WU, L.-S., MENCZER, F., AND MAGUITMAN, A. G. Emerging semantic communities in peer web search. In *Proceedings of the international workshop on Information retrieval in peer-to-peer networks* (New York, NY, USA, 2006), P2PIR '06, ACM, pp. 1–8.

En este trabajo se introducen dos criterios para evaluar el rendimiento de los métodos de búsqueda en sistemas P2P que se basan en el concepto de localidad semántica. Uno de los criterios está basado en la topología de la red resultante de las interacciones entre los pares y el otro en la similitud entre tópicos utilizada para seleccionar vecinos potencialmente útiles. En este artículo se incluyen simulaciones con distintos algoritmos de búsqueda en redes P2P utilizando un sistema llamado *6S*.

- [AY14] AROUR, K., AND YEFERNY, T. Learning model for efficient query routing in p2p information retrieval systems. *Peer-to-Peer Networking and Applications* (2014), 1–17.

En este artículo se presenta un modelo semántico para el ruteo de consultas en sistemas P2P. En este modelo, una consulta puede ser ruteada hacia determinados pares en lugar de propagarla hacia sus vecinos o en forma aleatoria. El principal objetivo de este modelo es obtener un mejor rendimiento en tareas no supervisadas mediante la incorporación de datos obtenidos mediante la experiencia adquirida de interacciones previas (historia pasada). Los autores presentan una implementación del modelo y una correspondiente evaluación.

- [B⁺08] BESNARD, P., ET AL. Towards argumentation-based contract negotiation. *Comput. Models Argument: Proc. COMMA 172* (2008), 134.

Los autores presentan un enfoque basado en la argumentación para la negociación entre agentes. Los contratos son vistos como transacciones abstractas de ítems entre un agente comprador y otro vendedor. Cada agente posee creencias, metas y preferencias que se utilizan en un proceso de negociación en dos fases. En este artículo se presenta un marco teórico para este enfoque y distintas técnicas de evaluación.

- [BA99] BARABÁSI, A.-L., AND ALBERT, R. Emergence of Scaling in Random Networks. *Science* 286, 5439 (1999), 509–512.

En este artículo se presenta un reporte respecto a la existencia de un alto grado de auto-organización en las grandes redes complejas. Se exploran distintos tipos de redes con esta característica, describiendo su topología y la forma en la que se relacionan y constituyen.

- [BB03] BARABÁSI, B. A.-L., AND BONABEAU, E. Scale-free. *Scientific American* 288, 5 (2003), 50–59.

En este artículo se presentan la definición y las principales propiedades de las redes libres de escala. Además, se presentan numerosos ejemplos y las métricas que se utilizan para detectar redes con estas propiedades.

- [BCAA04] BARBOSA, M. W., COSTA, M. M., ALMEIDA, J. M., AND ALMEIDA, V. A. F. Using locality of reference to improve performance of peer-to-peer applications. *SIGSOFT Software Engineering Notes* 29, 1 (January 2004), 216–227.

Los autores presentan una extensiva evaluación respecto al rendimiento de distintos algoritmos para la localización de contenido y recuperación de la información en sistemas P2P, en particular evalúan los sistemas *FreeNet* y *Gnutella*. Además proponen un nuevo algoritmo basado en localizar pares con intereses similares para localizar contenido en forma eficiente. Incluyen distintas simulaciones en entornos realistas y un extenso análisis de los resultados obtenidos.

- [BCD07] BENCH-CAPON, T. J., AND DUNNE, P. E. Argumentation in artificial intelligence. *Artificial intelligence* 171, 10-15 (2007), 619–641.

En este artículo se presentan una variedad de tendencias, desarrollos y aplicaciones que cubren un amplio rango de tópicos relacionados con la teoría y aplicaciones de la argumentación. Los autores presentan un contexto histórico respecto a los orígenes de la argumentación en el área de la Inteligencia Artificial e incluyen una discusión relacionada a la utilización de métodos basados en la argumentación en distintas áreas de aplicación.

- [BFR14] BABAEI, H., FATHY, M., AND ROMOOZI, M. Modeling and optimizing random walk content discovery protocol over mobile ad-hoc networks. *Performance Evaluation* 74 (2014), 18–29.

Los autores proponen un método adaptativo para optimizar el clásico algoritmo *Random Walk* de búsqueda en redes P2P no estructuradas. Además, presentan una evaluación en la que tienen en cuenta la cantidad de respuestas satisfactorias obtenidas y el tiempo que se tardó en encontrarlas. Por último consideraron el consumo de energía como un parámetro extra que es un factor importante en el rendimiento de los sistemas móviles.

- [BGSWW14] BEN-GAL, I., SHAVITT, Y., WEINSBERG, E., AND WEINSBERG, U. Peer-to-peer information retrieval using shared-content clustering. *Knowledge and information systems* 39, 2 (2014), 383–408.

En este artículo se presenta un estudio respecto a los clusters que se obtienen cuando se aplica algoritmos de clustering en grafos en los que el grado de sus nodos respeta una distribución de potencias. Además, se presenta un análisis de cómo estos clusters ayudan en la localización de contenido. Los autores proponen un algoritmo de clustering al que se analizó y comparó con el sistema *Gnutella*.

- [BM04] BRODER, A., AND MITZENMACHER, M. Network applications of bloom filters: A survey. *Internet mathematics* 1, 4 (2004), 485–509.

En este artículo se presenta una revisión de literatura respecto a la utilización de una estructura de datos denominada *Bloom filters* que es ampliamente usada en redes y bases de datos. El foco de este artículo se centra en aquellos sistemas que utilizan estas estructuras para resolver problemas en distintos tipos de redes.

- [BMNV13] BACCELLI, F., MATHIEU, F., NORROS, I., AND VARLOOT, R. Can p2p networks be super-scalable? In *INFOCOM, 2013 Proceedings IEEE* (2013), IEEE, pp. 1753–1761.

Se propone un modelo para redes P2P que toma en cuenta los cuellos de botella que impiden el acceso a los datos. Este modelo considera distintas cuestiones tales como el rendimiento en la descarga realizada por los usuarios. Los autores presentan un análisis de los resultados obtenidos donde se demuestra que el

promedio de tiempo de descarga baja a medida que aumenta la distribución de la carga, fenómeno que denominaron super-escalabilidad.

- [BMR03] BAWA, M., MANKU, G. S., AND RAGHAVAN, P. Sets: search enhanced by topic segmentation. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* (2003), ACM, pp. 306–313.

Los autores proponen en este artículo una arquitectura para búsquedas eficientes en redes P2P denominado *SETS*. Este enfoque combina ideas de machine learning con teoría de redes sociales. La idea es conectar pares con contenido similar y rutear las consultas sólo a aquellos nodos que se encuentran en regiones cercanas. Se presenta una discusión respecto a distintas cuestiones de diseño y cuestiones que se deberán tener en cuenta a la hora de la implementación de dicho sistema.

- [Bru46] BRUIJN, D. N. A combinatorial problem. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen. Series A* 49, 7 (1946), 758.

- [BTK93] BONDARENKO, A., TONI, F., AND KOWALSKI, R. A. An assumption-based framework for non-monotonic reasoning. In *LPNMR* (1993), vol. 93, pp. 171–189.

En este artículo se presentan aplicaciones de un framework basado en presunciones. Estas aplicaciones son la programación en lógica, programación lógica con negación clásica y lógica modal no monótona.

- [bud17] Buddycloud. <http://buddycloud.com/>. Recuperado en Junio de 2017, 2017.

BuddyCloud es un proyecto de código abierto que ayuda a los desarrolladores a añadir funciones de chat y video a su aplicación sin caer en una solución propietaria. *BuddyCloud* ofrece un servidor de mensajería de código abierto y trabajo con comités de estándares para asegurar que todo lo que se ofrece sea abierto y libre para que otros lo usen.

- [Bur92] BURT, R. S. *Structural holes: The social structure of competition*. Harvard University Press, 1992.

En este libro, su autor Ronald Burt describe la teoría de la estructura social de la competencia que se ha desarrollado durante las últimas dos décadas. El contraste entre la competencia perfecta y el monopolio se sustituye por un modelo de competencia en red. El elemento básico en este relato es el agujero estructural: una brecha entre dos individuos con recursos o información complementarios. Cuando los dos se conectan a través de un tercer individuo el empresario llena la brecha, creando importantes ventajas para el mismo. La ventaja competitiva es una cuestión de acceso a los agujeros estructurales en relación con las transacciones del mercado.

- [CDE08] CZENKO, M., DOUMEN, J., AND ETALLE, S. Trust management in p2p systems using standard tulip. In *IFIP International Conference on Trust Management* (2008), Springer, pp. 1–16.

En este artículo se presenta *Standard TuLip* que es un sistema lógico para el manejo de la confiabilidad. En este sistema las decisiones están basadas en credenciales de seguridad, que pueden ser demandadas por diferentes entidades y almacenadas en distintas locaciones. Los autores presentan el lenguaje de este sistema y las cuestiones prácticas para el desarrollo del mismo tales como la codificación de credenciales, la arquitectura del sistema, sus componentes, su funcionalidad, y también problemas de usabilidad.

- [CGH05] CHEN, H., GONG, Z., AND HUANG, Z. Self-learning routing in unstructured p2p network. *International journal of information technology* 11, 12 (2005), 59–67.

En este artículo se presenta un algoritmo denominado *SLPS* para la localización de contenidos en sistemas P2P. Este sistema posee un método de aprendizaje mediante el cual se toman en cuenta los intereses de los nodos a medida que transcurren interacciones entre los mismos. De este modo, es capaz de construir relaciones de amistad que pueden ser utilizadas para localizar el contenido en forma eficiente. Presentan una serie de resultados experimentales que muestran que *SLPS* es mejor en rendimiento respecto al algoritmo básico de inundación o *Flooding*.

- [CGM02] CRESPO, A., AND GARCIA-MOLINA, H. Routing indices for peer-to-peer systems. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on* (2002), IEEE, pp. 23–32.

Los autores presentan en este artículo el concepto de *Routing Indices*, que les permite a los nodos en un sistema P2P re-enviar sus consultas hacia vecinos que es probable que tengan una respuesta. Si un nodo no puede responder la consulta, la re-envía a un subconjunto de sus vecinos utilizando su índice de ruteo local en lugar de seleccionarlos al azar o inundando la red. Además, se presentan numerosos resultados y evaluaciones de los mismos.

- [CGM05] CRESPO, A., AND GARCIA-MOLINA, H. Semantic overlay networks for p2p systems. In *Agents and Peer-to-Peer Computing*. Springer, 2005, pp. 1–13.

En este trabajo la propuesta de los autores es atacar el problema de localización de contenidos en redes P2P mediante conexiones entre nodos basadas en su contenido, de esta manera se conectarían aquellos nodos con intereses similares. Bajo este modelo, las consultas se rutean hacia nodos específicos aumentando la probabilidad de encontrar una respuesta satisfactoria.

- [CH09] CRASWELL, N., AND HAWKING, D. Web information retrieval. *Information retrieval: Searching in the 21st century* (2009), 85–101.

- [CHC⁺11] CHOI, J., HAN, J., CHO, E., KWON, T. T., AND CHOI, Y. A survey on content-oriented networking for efficient content delivery. *Communications Magazine, IEEE* 49, 3 (2011), 121–127.

En este artículo se presenta una revisión del estado del arte respecto a algoritmos de ruteo en redes orientadas al contenido (CON), junto con una discusión respecto a líneas investigativas planteadas a futuro.

- [CKU09] CIRACI, S., KÖRPEOĞLU, I., AND ULUSOY, O. Reducing query overhead through route learning in unstructured peer-to-peer network. *J. Netw. Comput. Appl.* 32, 3 (May 2009), 550–567.

En este artículo se presenta el algoritmo de ruteo *Route Learning* cuya finalidad es reducir el tráfico de consultas en redes P2P no estructuradas. En este algoritmo cada nodo trata de identificar pares candidatos a satisfacer sus consultas. De esta manera, las consultas se envían sólo a un subconjunto de nodos de la

red. Se presentan resultados experimentales que se comparan con un algoritmo básico estilo *Flooding*.

- [CM05] CASTANO, S., AND MONTANELLI, S. Semantic self-formation of communities of peers. In *Proc. of the ESWC Workshop on Ontologies in Peer-to-Peer Communities, Heraklion, Greece* (2005).

Los autores de este artículo proponen un enfoque para la auto-organización de comunidades autónomas de pares en redes P2P. Se proponen técnicas basadas en unir y separar comunidades semánticas para luego aplicar algoritmos de búsqueda que exploten la formación de las mismas.

- [CMG09] CHESNEVAR, C. I., MAGUITMAN, A. G., AND GONZÁLEZ, M. P. Empowering recommendation technologies through argumentation., 2009.

En este capítulo se presenta un enfoque genérico para caracterizar un sistema recomendador basado en argumentos. Este enfoque se basa en modelar las preferencias de los usuarios en hechos, reglas estrictas y reglas rebatibles en el marco de un formalismo argumentativo. Estas preferencias se combinan con información adicional que es utilizada por el sistema argumentativo para priorizar sugerencias que serán el resultado final que se provee al usuario.

- [CML00] CHESÑEVAR, C. I., MAGUITMAN, A. G., AND LOUI, R. P. Logical models of argument. *ACM Computing Surveys (CSUR)* 32, 4 (2000), 337–383.

Los autores presentan un trabajo de revisión de literatura donde se discuten las principales ideas que caracterizan a diferentes modelos lógicos de argumentación. Además, se incluye la definición formal de las características de los principales enfoques para el modelado de la argumentación. Como característica a destacar, se incluye una revisión histórica de la argumentación desde mediados de los años ochenta, cuando los sistemas de argumentación surgían como una opción a los formalismos basados en lógica clásica.

- [CRL00] CARBOGIM, D. V., ROBERTSON, D., AND LEE, J. Argument-based applications to knowledge engineering. *The Knowledge Engineering Review* 15, 2 (2000), 119–149.

En este artículo se presentan diferentes cuestiones que pueden ser atacadas mediante el uso de sistemas argumentativos automatizados y se destacan im-

portantes direcciones de investigación en lo que respecta a ingeniería del conocimiento.

- [CSWH01] CLARKE, I., SANDBERG, O., WILEY, B., AND HONG, T. W. Freenet: A distributed anonymous information storage and retrieval system. In *Designing Privacy Enhancing Technologies* (2001), Springer, pp. 46–66.

En este artículo se presenta *FreeNet*, una aplicación P2P que permite la publicación y recuperación de datos protegiendo la identidad tanto de los autores como de los lectores. Este enfoque opera como una red de nodos idénticos que utilizan colectivamente su capacidad de almacenamiento y cooperan entre sí para rutear las consultas adecuadamente.

- [DC98] DU, A., AND CALLAN, J. Probing a collection to discover its language model. Tech. rep., 1998.

- [DGMY03] DASWANI, N., GARCIA-MOLINA, H., AND YANG, B. Open problems in data-sharing peer-to-peer systems. In *Database Theory—ICDT 2003*. Springer, 2003, pp. 1–15.

En este artículo se presentan y discuten futuras líneas investigativas para los mecanismos de búsquedas en sistemas P2P. Además, se hace énfasis en problemas que hasta ese momento no habían sido estudiados con demasiada profundidad. En particular, los autores hacen foco en dos aspectos principales de los sistemas P2P, por un lado la seguridad y por el otro los mecanismos de búsqueda.

- [dia17] Diaspora. <https://diasporafoundation.org/>. Recuperado en Junio, 2017.

Diaspora es una red social distribuida sin fines de lucro. El proyecto fue fundado en el año 2010 por cuatro estudiantes del Instituto de Courant de Ciencias Matemáticas de la Universidad de Nueva York. *Diaspora* consiste en un grupo de servidores independientes que interactúan entre sí para formar la red *Diaspora*. Cada servidor opera con una copia del software *Diaspora*, actuando como un servidor web personal. Los usuarios de la red pueden crear una cuenta en cualquier servidor que sea de su elección, pero aun así pueden interactuar con usuarios que se encuentran en otros servidores.

- [DKT06] DUNG, P. M., KOWALSKI, R. A., AND TONI, F. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence* 170, 2 (2006), 114–159.

Los autores presentan una familia de pruebas dialécticas para la admisibilidad de semánticas en el framework *assumption-based argumentation*. Estos procedimientos de prueba se definen para cualquier lógica convencional formulada como una colección de reglas de inferencia y se muestra cómo tal lógica puede extenderse a un sistema de argumentación dialéctica. Los procedimientos de prueba comparten la misma noción de estrategia ganadora para una disputa y difieren sólo en la estrategia de búsqueda que utilizan para encontrarla. La novedad del enfoque que se propone radica principalmente en el uso del razonamiento hacia atrás para construir argumentos y el hecho de que el proponente y el oponente pueden atacarse unos a otros antes de que se complete un argumento.

- [DKT09] DUNG, P. M., KOWALSKI, R. A., AND TONI, F. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*. Springer, 2009, pp. 199–218.

En este capítulo se define el framework *Assumption-based argumentation* (ABA). Se presentan las principales características de ABA respecto a las nociones de argumentos y ataques. Además, se hace énfasis en la noción de aceptabilidad de un argumento o de un conjunto de argumentos bajo distintas semánticas. Finalmente presentan cómo es que se computa el conjunto de argumentos ganadores y cómo es que todos estos conceptos se aplican en problemáticas de la vida real.

- [DMT07] DUNG, P. M., MANCARELLA, P., AND TONI, F. Computing ideal sceptical argumentation. *Artificial Intelligence* 171, 10 (2007), 642–674.

En este artículo se presentan dos procedimientos dialécticos para la semántica ideal escéptica en la argumentación. En el primer caso el procedimiento se define en términos de árboles de disputa para frameworks de argumentación abstracta. En el segundo caso se define un procedimiento en términos dialécticos para la argumentación basada en presunciones (*assumption-based argumentation*).

- [DNT02] DIMOPOULOS, Y., NEBEL, B., AND TONI, F. On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence* 141, 1-2 (2002), 57–78.
- En este artículo se analiza la complejidad computacional del razonamiento crédulo y escéptico bajo la semántica de argumentos admisibles y preferidos para la variante proposicional de las instancias de un framework abstracto que capturan teoría, circunscripción, programación lógica y lógica por defecto.
- [Dun95] DUNG, P. M. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence* 77, 2 (1995), 321–357.
- Los autores realizan un estudio respecto a cómo es que los seres humanos utilizan la argumentación y cómo podría implementarse este mecanismo en las computadoras. Además, se presenta una implementación basada en la programación en lógica para la generación de meta-intérpretes..
- [Dun09] DUNNE, P. E. The computational complexity of ideal semantics. *Artificial Intelligence* 173, 18 (2009), 1559–1591.
- En este artículo se presenta un análisis de la complejidad computacional de la semántica ideal tanto en la argumentación abstracta como en la argumentación basada en presunciones (assumption-based argumentation).
- [DZGL05] DUNN, R. J., ZAHORJAN, J., GRIBBLE, S. D., AND LEVY, H. M. Presence-based availability and p2p systems. In *Peer-to-Peer Computing, 2005. P2P 2005. Fifth IEEE International Conference on* (2005), IEEE, pp. 209–216.
- En este artículo se presenta una métrica llamada *presence-based availability* para calcular la disponibilidad de un servicio P2P que toma en cuenta las disponibilidades individuales de los nodos. Demuestran con distintos ejemplos que esta métrica es un indicador confiable de la disponibilidad.
- [EB05] EVERETT, M., AND BORGATTI, S. P. Ego network betweenness. *Social Networks* 27, 1 (2005), 31–38.
- [ECMB12] ESTEVEZ, E., CHESÑEVAR, C., MAGUITMAN, A., AND BRENA, R. Decide 2.0: a framework for intelligent processing of citizens' opinion in social

media. In *Proceedings of the 13th Annual International Conference on Digital Government Research* (2012), ACM, pp. 266–267.

En este artículo se presenta el proyecto *DECIDE 2.0* que incluye un framework y una herramienta de software que les permite a los ciudadanos expresar sus opiniones en forma digital. En este proyecto se combina el concepto de búsqueda basada en contexto y argumentación en un sistema colaborativo para el manejo de información relacionada a cuestiones gubernamentales.

- [ER59] ERDŐS, P., AND RÉNYI, A. On random graphs. *Publicationes Mathematicae Debrecen* 6 (1959), 290–297.

En este artículo se presenta un método para la generación de grafos aleatorios. En este modelo un nuevo nodo se enlaza con igual probabilidad con el resto de la red, es decir, posee una independencia estadística con el resto de los nodos de la red. Hoy en día es la base teórica para la generación de otras redes.

- [FGT96] FASS, C., GINELLI, M., AND TURTLE, B. *Six degrees of kevin bacon*. Plume Books, 1996.

En este libro se presenta *Six Degrees de Kevin Bacon* que es un juego basado en el concepto de *seis grados de separación*, que postula que dos personas en la Tierra están separados por seis o menos vínculos de amistad entre ellos. Los participantes se desafían mutuamente para encontrar el camino más corto entre un actor arbitrario y el actor Kevin Bacon. Se basa en el supuesto de que cualquier persona involucrada en la industria cinematográfica de Hollywood puede estar vinculada a través de sus papeles cinematográficos a Bacon en seis pasos. El juego requiere de un grupo de jugadores para tratar de conectar a cualquier individuo con Kevin Bacon lo más rápido posible y en el menor número de enlaces.

- [FRE11] Freenet, 2011.

Freenet es una red de distribución de información descentralizada y resistente a la censura. Su objeto es proporcionar libertad de expresión a través de las redes P2P mediante la protección del anonimato. Esta herramienta es de software libre y actualmente se encuentran versiones completamente funcionales.

- [fri17] Friendica. <http://friendi.ca/>. Recuperado en Junio 2017, 2017.

Friendica es una red social distribuida. Los servidores que ejecutan *Friendica* son compatibles con *Diaspora* y *Hubzilla*. Esta red social es desarrollada por muchas personas diferentes de todo el mundo. No hay corporación ni ninguna fundación detrás de *Friendica*.

- [FV17] FANTI, G., AND VISWANATH, P. Anonymity properties of the bitcoin p2p network. *arXiv preprint arXiv:1703.08761* (2017).

En este artículo se presenta un modelo del sistema *Bitcoin* mediante el cual se pretende analizar sus propiedades respecto al anonimato, tanto cuando el sistema utilizaba el algoritmo *Flooding* como actualmente que utiliza un método llamado *difusión*.

- [GCGMP97] GRAVANO, L., CHANG, K., GARCIA-MOLINA, H., AND PAEPCKE, A. Starts: Stanford protocol proposal for internet retrieval and search. Tech. rep., Stanford, CA, USA, 1997.

Los autores presentan un protocolo para Internet denominado *STARS*. Este protocolo busca brindar facilidades al momento de realizar búsquedas en distintas fuentes. En este artículo se presenta el protocolo propuesto y además se incluye una discusión respecto a cómo es que surgió la idea para desarrollar el mismo.

- [GMS04] GKANTSIDIS, C., MIHAIL, M., AND SABERI, A. Random walks in peer-to-peer networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies* (2004), vol. 1, IEEE.

En este artículo se presenta un exhaustivo análisis sobre el algoritmo *Random-walks*. Las simulaciones fueron realizadas sobre redes P2P no estructuradas y se evaluaron distintos escenarios. Los autores muestran los resultados obtenidos y una discusión respecto a los mismos.

- [GNU17] Gnusocial. <https://gnu.io/social/>. Recuperado en Junio, 2017.

GNUSocial es una continuación del proyecto *StatusNet*. Es un software de comunicación social, tanto para comunicaciones públicas como privadas. Provee una alternativa para aquellos usuarios que quieren abandonar los servicios capitalistas centralizados.

- [GS03] GARCIA, A. J., AND SIMARI, G. R. Defeasible logic programming: An argumentative approach. *arXiv preprint cs/0302029* (2003).

En este artículo sus autores introducen *Defeasible Logic Programming* (DeLP), un formalismo que combina resultados de programación en lógica y argumentación rebatible. En este formalismo la argumentación se utiliza para decidir entre metas contradictorias. Las consultas están soportadas por argumentos que pueden ser derrotados por otros argumentos. DeLP se puede utilizar para construir aplicaciones que necesiten manejar información inconsistente y contradictoria en dominios dinámicos. Por ejemplo, se puede utilizar para representar el conocimiento de un agente.

- [GS04] GARCÍA, A. J., AND SIMARI, G. R. Defeasible logic programming: An argumentative approach. *Theory Pract. Log. Program.* 4, 2 (January 2004), 95–138.
 - [GS09] GÜRSEL, A., AND SEN, S. Producing timely recommendations from social networks through targeted search. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2* (2009), International Foundation for Autonomous Agents and Multiagent Systems, pp. 805–812.
 - [HB15] HADDI, F. L., AND BENCHAIÏBA, M. A survey of incentive mechanisms in static and mobile p2p systems. *Journal of Network and Computer Applications* 58 (2015), 108–118.
- En este artículo se presentan las principales técnicas de incentivos utilizadas en sistemas P2P. Los incentivos son una forma de motivar a los pares a colaborar simultáneamente para alcanzar determinados objetivos. Múltiples mecanismos de incentivos han sido desarrollados con el objetivo de enriquecer la colaboración entre los pares. Además, los autores presentan una novedosa clasificación de estos mecanismos.
- [HCHL05] HUANG, X., CHEN, L., HUANG, L., AND LI, M. Routing algorithm using skipnet and small-world for peer-to-peer system. *Lecture notes in computer science* 3795 (2005), 984.
 - [Her05] HERSCHEL, S. Indexing dynamic networks. In *GI Jahrestagung (1)* (2005), pp. 429–433.

En este artículo se presenta una arquitectura para el procesamiento de consultas en redes P2P llamado *The Query Web*. Este es un sistema híbrido que combina un enfoque de tablas hash distribuidas con redes P2P no estructuradas.

- [Hua08] HUANG, A. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, Christchurch, New Zealand (2008), pp. 49–56.

En este artículo se presenta un análisis del rendimiento de distintas métricas que se utilizan para el cálculo de distancia y similitud en los algoritmos de clustering.

- [HWS13] HSU, C.-Y., WANG, K., AND SHIH, H.-C. Decentralized structured peer-to-peer network and load balancing methods thereof, May 14 2013. US Patent 8,443,086.

- [JNCY06] JIN, H., NING, X., CHEN, H., AND YIN, Z. Efficient query routing for information retrieval in semantic overlays. In *In Proc. of the 21st Annual ACM Symposium on Applied Computing (SAC'06)* (2006), ACM Press, pp. 23–27.

En este artículo se propone un modelo en el cual los pares de una red P2P semánticamente similares son agrupados para luego lograr un envío eficiente de consultas. Los autores presentan distintos resultados en donde se puede determinar que el método funciona mejor respecto a otros que no tienen en cuenta la semántica de los datos ni los intereses de los usuarios.

- [Jos02] JOSEPH, S. NeuroGrid: Semantically Routing Queries in Peer-to-Peer Networks. In *Web Engineering and Peer-to-Peer Computing*, E. Gregori, L. Cherkasova, G. Cugola, F. Panzieri, and G. Picco, Eds., vol. 2376 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2002, pp. 202–214.

En este artículo se presenta un sistema de búsqueda descentralizado llamado *NeuroGrid*. Cada nodo de este sistema soporta un algoritmo de búsqueda distribuida mediante un ruteo semántico, es decir, que las consultas se rutean según su contenido. Además, este sistema posee un método de aprendizaje dinámico que captura meta-datos que describen el contenido de los nodos y de sus archivos.

- [jun] Jung. <http://jung.sourceforge.net> - Recuperado en Marzo del 2014.
- Jung es una herramienta para la visualización de grafos escrita en Java. Esta herramienta permite modelar una gran cantidad de representaciones de grafos y ofrece distintas opciones a la hora de visualizarlos.
- [JW04] JAWHAR, I., AND WU, J. A two-level random walk search protocol for peer-to-peer networks. In *Proc. of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics* (2004), pp. 1–5.
- En este artículo se presenta una estrategia de búsqueda en redes P2P denominada *two-level random walk*. Este método trata de bajar el número de mensajes redundantes que circula por la red sin penalizar significativamente el tiempo que se tarda en encontrar una respuesta. Los autores presentan las principales definiciones que intervienen en este enfoque junto a resultados empíricos.
- [KaZ] Kazaa.
- KaZaA fue una aplicación P2P para el intercambio de archivos. Este sistema era utilizado para el intercambio de música (principalmente en formato MP3) y películas (en formato DivX). Este sistema estuvo involucrado en distintos problemas legales relacionados con los derechos de autor. Desde agosto de 2012 el sitio web de KaZaA fue desactivado definitivamente.
- [KGZY02a] KALOGERAKI, V., GUNOPULOS, D., AND ZEINALIPOUR-YAZTI, D. A local search mechanism for peer-to-peer networks. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management* (New York, NY, USA, 2002), CIKM '02, ACM, pp. 300–307.
- [KGZY02b] KALOGERAKI, V., GUNOPULOS, D., AND ZEINALIPOUR-YAZTI, D. A local search mechanism for peer-to-peer networks. In *Proceedings of the eleventh international conference on Information and knowledge management* (2002), ACM, pp. 300–307.
- En este artículo se proponen dos métodos para la recuperación de información en redes P2P puras. En primer lugar se propone una modificación del algoritmo *Breadth-First-Search* (BFS) y en segundo lugar un mecanismo de búsqueda inteligente íntegramente distribuido. Se presentan distintas simulaciones y resultados obtenidos a partir de las mismas.

- [KJ07] KLAMPANOS, I., AND JOSE, J. An evaluation of a cluster-based architecture for peer-to-peer information retrieval. *Lecture Notes in Computer Science* 4653 (2007), 380–391.
- En este artículo se presenta una evaluación de una arquitectura basada en clustering para la recuperación de información en sistemas P2P.
- [Kle06] KLEINBERG, J. Complex networks and decentralized search algorithms. In *Proceedings of the International Congress of Mathematicians (ICM)* (2006), vol. 3, pp. 1019–1044.
- En este artículo se presenta una revisión de literatura respecto al estudio de las redes complejas. En particular el trabajo se centra en el fenómeno conocido como *small-world* y en los algoritmos de búsqueda distribuidos. Se presenta y ejemplifica el experimento popularmente conocido llamado *seis grados de separación* en redes sociales y luego se presenta una discusión respecto a algunos modelos probabilísticos surgidos a partir del estudio de las redes complejas.
- [KR02] KUROSE, J. F., AND ROSS, K. *Computer Networking: A Top-Down Approach Featuring the Internet*, 2nd ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
- En este libro que presentan y definen los principales conceptos involucrados en redes de computadoras. Utiliza un enfoque Top-Down, lo que significa que comienza estudiando aplicaciones de redes para luego llegar a la capa mas baja donde se refiere a redes a nivel físico.
- [Kre04] KRETSCHMER, H. Author productivity and geodesic distance in bibliographic co-authorship networks, and visibility on the web. *Scientometrics* 60, 3 (2004), 409–420.
- [kun] Kune. <http://kune.ourproject.org>. Recuperado en Junio 2017.
- Kune* es una nueva herramienta web, todavía en desarrollo, para la creación de ambientes de intercomunicación permanente, inteligencia colectiva, conocimiento y trabajo compartido. *Kune* tiene como principal objetivo multiplicar los potenciales de las personas, comunidades y organizaciones, permitiendo una cooperación continua, sin tener en cuenta la distancia física. Así, permitirá

realizar reuniones virtuales, coordinar agendas comunes, unir a personas con intereses comunes y desarrollar proyectos cooperativos.

- [KXZ05] KUMAR, A., XU, J., AND ZEGURA, E. W. Efficient and scalable query routing for unstructured peer-to-peer networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* (2005), vol. 2, IEEE, pp. 1162–1173.

En este artículo se propone un protocolo para el ruteo de consultas que garantiza un bajo uso del ancho de banda. Para lograr esto, este enfoque utiliza tablas de ruteo probabilísticas que son almacenadas en cada nodo. Sin embargo, para mantener estas tablas actualizadas se necesitan mecanismos no tradicionales. Con el objetivo de sobrellevar esta cuestión, los autores presentan una estructura de datos denominada *Exponentially Decaying Bloom Filter* (EDBF) que facilita la actualización del contenido de las tablas.

- [LC06] LU, J., AND CALLAN, J. Full-text federated search of text-based digital libraries in peer-to-peer networks. *Information Retrieval* 9, 4 (2006), 477–498.

En este artículo se presenta una solución al problema de la búsqueda de texto en librerías digitales basadas en redes P2P jerárquicas. Se presentan modificaciones de algoritmos ya existentes para el problema de la representación y selección de los recursos. Los resultados experimentales que se presentan demuestran una mejora respecto a los sistemas de búsqueda presentes en las librerías digitales.

- [LC07] LU, J., AND CALLAN, J. Content-based peer-to-peer network overlay for full-text federated search. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)* (Paris, France, France, 2007), RIAO '07, LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, pp. 490–509.

En este artículo se propone una red P2P de contenidos de texto. Se presentan algoritmos de búsqueda locales para construir dinámicamente una red lógica basada en la localidad del contenido. Se presentan distintos resultados experimentales junto con un análisis de los mismos.

- [LCC⁺02] LV, Q., CAO, P., COHEN, E., LI, K., AND SHENKER, S. Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th international conference on Supercomputing* (2002), ACM, pp. 84–95.
- En este artículo se proponen y discuten distintas alternativas al algoritmo *Flooding* para la localización de contenidos en redes P2P puras. En particular, el estudio se centra en un algoritmo basado en múltiples *random-walk* que según los resultados experimentales demuestra ser rápido y disminuye el tráfico en la red.
- [LCP⁺05] LUA, E. K., CROWCROFT, J., PIAS, M., SHARMA, R., AND LIM, S. A survey and comparison of peer-to-peer overlay network schemes. *Commun. Surveys Tuts.* 7, 2 (April 2005), 72–93.
- En este artículo se presenta una revisión del estado del arte en redes P2P tanto estructuradas como no estructuradas. Se presenta una clasificación de las mismas según su grado de centralización y se discute el rendimiento de las aplicaciones P2P en cada uno de estos grupos.
- [LKG13] LI, J., KHAN, S. U., AND GHANI, N. Semantics-based resource discovery in large-scale grids. *Large Scale Network-centric Computing Systems* (2013).
- Los autores presentan un trabajo de investigación respecto a la localización de recursos en redes de escala global. Además, se presenta un framework para la localización de contenido basado en la semántica de los datos. Se muestra cómo este framework puede ser utilizado para localizar contenidos en grandes redes. Se presentan distintos resultados que se analizan en base a la expresividad, escalabilidad y eficiencia.
- [LST05] LÖSER, A., STAAB, S., AND TEMPICH, C. Semantic methods for p2p query routing. In *Multiagent System Technologies*. Springer, 2005, pp. 15–26.
- En este artículo se presenta un estudio respecto a los asuntos o problemas presentes en sistemas de búsqueda descentralizados. Ejemplos de estas cuestiones son la independencia de los pares, el control sobre los recursos, métodos de ruteo, entre otros. Para sobrellevar estas cuestiones los autores utilizan un enfoque basado en metáforas sociales y similitud semántica entre consultas y

fuentes de información. Se presenta un análisis de los resultados obtenidos mediante simulaciones y se menciona una aplicación denominada *Bibster* que es una implementación parcial del sistema propuesto.

- [LW06] LI, X., AND WU, J. Searching techniques in peer-to-peer networks. *Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, and Peer-to-Peer Networks* (2006), 613–642.

- [LWAM09] LELE, N., WU, L.-S., AKAVIPAT, R., AND MENCZER, F. Sixearch.org 2.0 peer application for collaborative web search. In *Hypertext* (2009), pp. 333–334.

En este artículo se introduce *Sixearch.org*, una aplicación P2P para la búsqueda adaptativa de contenido. Esta aplicación está compuesta de un protocolo, un *web crawler*, un mecanismo de indexado de documentos y un sistema de aprendizaje contextual. En *Sixearch.org* cada usuario tiene su propio perfil. De esta manera, cuando genera una consulta, la misma es ruteada hacia pares potencialmente útiles en base a la historia pasada de ese perfil.

- [LYHC14] LI, R.-H., YU, J. X., HUANG, X., AND CHENG, H. Random-walk domination in large graphs. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on* (2014), IEEE, pp. 736–747.

En este artículo se presentan y formulan dos tipos de problemas de dominio en enfoques *random-walk* en grafos. Estos problemas fueron motivados por aplicaciones prácticas tal como el problema de la localización de ítems en redes sociales o en redes P2P. Para solucionar estos problemas se propone un algoritmo de programación dinámica basado en otro algoritmo codicioso. Se presenta un gran número de experimentos y se demuestra la eficiencia y escalabilidad de los algoritmos propuestos.

- [Mal15] MALATRAS, A. State-of-the-art survey on p2p overlay networks in pervasive computing environments. *Journal of Network and Computer Applications* 55 (2015), 1 – 23.

En este artículo se presenta un estudio respecto al estado del arte en sistemas P2P. En particular, este estudio se centra en cómo se construyen otras redes

lógicas sobre la red física que conforma el sistema. Se describen y comparan numerosas aplicaciones que trabajan con este tipo de redes intermedias.

- [mas17] Mastodon. <https://mastodon.social>. Retrieved in June 2017, 2017.
- Mastodon* en una red social distribuida similar a *Twitter*. Lo que la diferencia de esta última es su descentralización y su política de código abierto. Actualmente *Mastodon* se encuentra disponible para navegadores web y para dispositivos móviles.
- [MDPY13] MENG, F., DING, L., PENG, S., AND YUE, G. A p2p network model based on hierarchical interest clustering algorithm. *Journal of Software* 8, 5 (May 2013), 1262–1267.
- En este artículo se presenta un sistema jerárquico de clustering para mejorar la eficiencia de las búsquedas en redes P2P. Estas redes son divididas en diferentes sub-redes de acuerdo a los intereses de sus participantes, construyendo de este modo una topología estructural. Se muestran distintos resultados producto de simulaciones que demuestran una optimización en el uso de los recursos.
- [MGSN05] MICHLMAYR, E., GRAF, S., SIBERSKI, W., AND NEJDL, W. Query routing with ants.
- Los autores presentan *SemAnt*, un algoritmo diseñado para el ruteo de consultas en entornos P2P basado en la manera en la que las hormigas hallan sus senderos. Se presenta el comportamiento natural de las hormigas en la vida real y se muestra cómo algunas de sus conductas pueden ser aplicadas al ruteo de consultas.
- [min17] Minds. <https://www.minds.com/>. Retrieved in June 2017, 2017.
- Minds* es una red social distribuida cuyo objetivo principal es respetar la privacidad de sus usuarios. *Minds* es una alternativa a las redes sociales más populares y está enfocada para aquellos que busquen un entorno seguro y sobre todo gratuito. En este sistema todos los mensajes que se envían están encriptados. Además, el proyecto es de código abierto, lo que significa que cualquiera puede contribuir a mejorar el diseño o el funcionamiento de esta nueva red social.
- [Mor07] MORR, D. Lionshare: A federated p2p app. In *Internet2 members meeting* (2007).

Lionshare es un sistema P2P que ha demostrado su potencial para la aplicación de sistemas avanzados para el soporte de medios colaborativos. Esta tecnología se ha utilizado para enseñar, aprender e investigar en un contexto de intercambio de información. Además, *LionShare* proporciona herramientas para la organización automatizada de recursos personales, búsquedas seguras en redes P2P y en repositorios institucionales, publicación segura de materiales académicos para comunidades seleccionadas y un método para una colaboración segura entre colegas.

- [MP13] MODGIL, S., AND PRAKKEN, H. A general account of argumentation with preferences. *Artificial Intelligence* 195 (2013), 361–397.

En este artículo se presenta una modificación del formalismo *ASPIC+*, para la argumentación con preferencias. Se presenta y define el framework en cuestión junto con un estudio sobre distintas maneras de instanciarlo utilizando la lógica clásica.

- [MTSD08] MATT, P.-A., TONI, F., STOURNARAS, T., AND DIMITRELOS, D. Argumentation-based agents for eprocurement. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track* (2008), International Foundation for Autonomous Agents and Multiagent Systems, pp. 71–74.

- [Nah04] NAH, F. F.-H. A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology* 23, 3 (2004), 153–163.

En este artículo se estudia la tolerancia de los usuarios respecto a la espera que les provoca el uso de algún sistema computacional. En particular se estudia la tolerancia a la espera que producen los mecanismos de recuperación de información. El análisis de los resultados empíricos ha demostrado que la tolerancia a la espera por parte de los usuarios es de aproximadamente dos segundos antes de abandonar la búsqueda.

- [Nak] NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system.

Bitcoin es una criptomoneda creada en el año 2009. Este sistema es sustentado por una red P2P descentralizada, es decir, que *Bitcoin* no depende de ninguna

entidad central. De igual manera las transacciones no necesitan de ningún intermediario, ofreciéndole de este modo un alto grado de confidencialidad a sus usuarios. *Blockchain* le agrega una funcionalidad extra a *Bitcoin* permitiéndole emitir, monitorear y transferir representaciones virtuales de activos externos.

- [Nap] Napster. <http://free.napster.com>. Recuperado en 2011.
- Napster* es un servicio de distribución de archivos de música (en formato MP3). Fue la primera gran red P2P de intercambio de contenido creada por Sean Parker y Shawn Fanning. Su popularidad comenzó durante el año 2000. Su tecnología permitía a los aficionados a la música compartir sus colecciones de MP3 fácilmente con otros usuarios, lo que originó las protestas de las instituciones de protección de derechos de autor.
- [Nas03] NASCIMENTO, M. A. Peer-to-peer: Harnessing the power of disruptive technologies. *SIGMOD Rec.* 32, 2 (June 2003), 57–58.
- [NCR08] NGUYEN, G. H., CHATALIC, P., AND ROUSSET, M.-C. A probabilistic trust model for semantic peer to peer systems. In *Proceedings of the 2008 international workshop on Data management in peer-to-peer systems* (2008), ACM, pp. 59–65.
- En este artículo se presenta un modelo probabilístico para manejar la confiabilidad en sistemas P2P. Consiste en un método local para computar la confiabilidad junto con un mecanismo sencillo de propagación de esta información.
- [NLMC13] NICOLINI, A. L., LORENZETTI, C. M., MAGUITMAN, A. G., AND CHESÑEVAR, C. I. Intelligent algorithms for reducing query propagation in thematic p2p search. In *Anales del XIX Congreso Argentino de Ciencias de la Computación (CACIC)* (Mar del Plata, Buenos Aires, Argentina, October 2013), pp. 71–79.
- En este artículo se presentan distintos algoritmos para reducir la propagación de consultas en redes P2P. Además, se presentan algunos resultados preliminares obtenidos mediante simulaciones ejecutadas sobre una red de mil participantes.
- [NLMC17] NICOLINI, A. L., LORENZETTI, C. M., MAGUITMAN, A. G., AND CHESÑEVAR, C. I. Intelligent algorithms for improving communication

patterns in thematic p2p search. *Information Processing and Management* 53, 2 (2017), 388–404.

En este artículo se presenta una serie de algoritmos cuya finalidad es reducir la propagación de consultas en redes P2P. Estos algoritmos se presentan de manera incremental. Esto es, desde un algoritmo básico que carece de inteligencia hasta uno que utiliza un razonamiento avanzado para decidir cómo rutear una consulta. Se presentan resultados de numerosas simulaciones donde se evaluó tanto el rendimiento de los algoritmos respecto a la cantidad de mensajes generados como la escalabilidad de los mismos. Además, a partir de los resultados se puede observar el surgimiento de una topología lógica de mundo pequeño que es evidencia del uso de buenos patrones de comunicación.

- [NMC15] NICOLINI, A. L., MAGUITMAN, A. G., AND CHESÑEVAR, C. I. Argp2p: An argumentative approach for intelligent query routing in P2P networks. In *Theory and Applications of Formal Argumentation - Third International Workshop, TAFE 2015, Buenos Aires, Argentina, July 25-26, 2015, Revised Selected Papers* (2015), pp. 194–210.

En este artículo se presenta un enfoque para el ruteo de consultas en redes P2P que utiliza argumentación para la toma de decisiones. El sistema propuesto se denomina *ArgP2P* y utiliza un algoritmo imperativo para la implementación de un sistema de búsqueda temática combinado con un sistema argumentativo. Este último sistema permite razonar respecto a cuándo es momento de explorar la red para ampliar el conocimiento global del sistema. Se presenta una formalización del enfoque propuesto junto con un caso de estudio en el que se puede observar el comportamiento del sistema propuesto bajo distintas circunstancias.

- [OU11] OKUBO, T., AND UEDA, K. Peer-to-peer contents delivery system considering network distance. In *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific* (2011), IEEE, pp. 1–4.

En este artículo se presenta un sistema para compartir contenido en redes P2P que considera la distancia física entre los pares mediante el uso del parámetro *time-to-live* (TTL). En este sistema los pares priorizan a los vecinos más cercanos para la recuperación de archivos. De esta manera se forman *clusters* basados en la distancia entre los pares. Se presentan resultados producto de

simulaciones que indican que este sistema reduce el tráfico de mensajes a lo largo de toda la red.

- [Pas12] PASSARELLA, A. A survey on content-centric technologies for the current internet: Cdn and p2p solutions. *Computer Communications* 35, 1 (2012), 1–32.

En este artículo se presenta una revisión de la literatura respecto a cómo es que Internet es una red tan dinámica y flexible. Se describen en detalle dos de las tecnologías mas populares: las redes de distribución de contenido y los sistemas P2P. Para cada uno de ellas, se muestran sus principales componentes y se discuten los distintos trabajos que se encuentran en la literatura sobre estas tecnologías.

- [Pol87] POLLOCK, J. L. Defeasible reasoning. *Cognitive science* 11, 4 (1987), 481–518.

En este artículo se presentan las principales características y definiciones involucradas en el concepto de razonamiento por defecto. Además se discute cómo un programa computacional puede implementar esta teoría.

- [Pon93] PONGOR, G. Omnet: Objective modular network testbed. In *Proceedings of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems* (San Diego, CA, USA, 1993), MASCOTS '93, Society for Computer Simulation International, pp. 323–326.

El autor de este artículo presenta distintos usos de la herramienta *OMNeT*. Esta misma permite la simulación de una gran cantidad de redes que el usuario puede definir respecto a su topología y comportamiento. En particular se presentan distintas aplicaciones que utilizaron esta herramienta para su etapa de simulación.

- [QMB⁺16] QAMAR, M., MALIK, M., BATOOL, S., MEHMOOD, S., MALIK, A. W., AND RAHMAN, A. Centralized to decentralized social networks: Factors that matter. *Managing and Processing Big Data in Cloud Computing* (2016), 37.

En este artículo se presentan investigaciones realizadas respecto a la descentralización de las redes sociales. También se presenta un estudio respecto a redes sociales centralizadas y posibles formas de descentralizarlas. Dentro de estas soluciones se presenta el uso de las redes P2P como una alternativa que ha demostrado tener un buen rendimiento.

- [QYL14] QIN, C., YANG, Z., AND LIU, H. User interest modeling for p2p document sharing systems based on k-medoids clustering algorithm. In *Computational Sciences and Optimization (CSO), 2014 Seventh International Joint Conference on* (2014), IEEE, pp. 576–578.

En este artículo se presenta un método para el modelado de intereses de usuarios en sistemas P2P. Este método está basado en el algoritmo de clustering *K-medoids*. El enfoque que proponen los autores crea un modelo inicial sobre el interés de los usuarios mediante el algoritmo de clustering. Luego, se combina con los resultados obtenidos por el usuario en el pasado para enriquecer el modelo.

- [RD01] ROWSTRON, A., AND DRUSCHEL, P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware 2001* (2001), Springer, pp. 329–350.

En este artículo se presenta el diseño y evaluación de *Pastry*, un sistema escalable y distribuido para la localización y ruteo de contenido en aplicaciones P2P. Este método se basa en designar identificadores a los nodos de una manera determinada y luego rutear las consultas a los nodos que poseen identificadores numéricamente más cercanos. Este sistema es descentralizado y auto-organizado. Además, mediante simulaciones ha demostrado ser escalable y se adapta a fallos de nodos.

- [Rei80] REITER, R. A logic for default reasoning. *Artificial intelligence* 13, 1-2 (1980), 81–132.

- [RFH⁺01] RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SHENKER, S. *A scalable content-addressable network*, vol. 31. ACM, 2001.

Los autores presentan el concepto de *Content-Addressable Network* (CAN), una infraestructura distribuida que utiliza tablas hash distribuidas. Se presenta su

definición y principales componentes y métodos utilizados. Además, se incluyen numerosos resultados que han demostrado la escalabilidad y la resistencia a fallos presentada por el sistema propuesto.

- [RGKK08] ROSENFELD, A., GOLDMAN, C. V., KAMINKA, G. A., AND KRAUS, S. Phirst: A distributed architecture for p2p information retrieval, 2008.

En este artículo se presenta *PHIRST*, un sistema para facilitar la búsqueda efectiva de texto en bases de datos P2P. Este sistema combina un enfoque centralizado para la publicación de los términos claves del contenido de los pares junto con un enfoque descentralizado para su mecanismo de búsqueda. Se evaluó el sistema en cuanto a su rendimiento en las búsquedas y en cuanto a su tolerancia a fallos.

- [Rip01] RIPEANU, M. Peer-to-peer architecture case study: Gnutella network. In *Proceedings of the First International Conference on Peer-to-Peer Computing* (2001), pp. 99–100.

En este artículo se analiza la topología y el tráfico en *Gnutella*. Se comprueba que este sistema no hace un uso eficiente de la topología física de la red, por lo tanto se proponen algunos cambios al protocolo empleado por *Gnutella* para solucionar esta cuestión.

- [RIS⁺94] RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P., AND RIEDL, J. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (1994), ACM, pp. 175–186.

En este artículo se presenta un sistema denominado *GroupLens* para el filtrado colaborativo de noticias web. Este sistema ayuda a las personas a encontrar artículos que les sean de su interés entre una gran cantidad de artículos que se encuentran disponibles en la red.

- [RM06] RISSON, J., AND MOORS, T. Survey of research towards robust peer-to-peer networks: search methods. *Computer networks* 50, 17 (2006), 3485–3521.

En este artículo se presenta una revisión de literatura respecto a algoritmos de ruteo en redes P2P. Los algoritmos de ruteo son una de las componentes

más importantes de un sistema P2P ya que son los que permiten localizar el contenido. Los autores describen distintos algoritmos de búsqueda teniendo en cuenta qué tanto ayudan a hacer el sistema más robusto..

- [SCG⁺16] SHEN, X.-J., CHANG, Q., GOU, J.-P., MAO, Q.-R., ZHA, Z.-J., AND LU, K. Collaborative q-learning based routing control in unstructured p2p networks. In *MultiMedia Modeling* (2016), Springer, pp. 910–921.

- [Sch01] SCHOLLMEIER, R. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Proceedings of the First International Conference on Peer-to-Peer Computing* (Washington, DC, USA, 2001), P2P '01, IEEE Computer Society, pp. 101–.

En este artículo se presenta una definición sobre redes P2P. Además, se incluye una clasificación de conceptos vinculados a Internet y a redes P2P tales como sistemas P2P puros, híbridos, modelo cliente-servidor, etc.

- [SE00] SUGIURA, A., AND ETZIONI, O. Query routing for web search engines: Architecture and experiments. *Computer Networks* 33, 1 (2000), 417–429.

En este artículo se presenta un sistema automático para el ruteo de consultas denominado *Q-Pilot*. Este sistema cuenta con una componente fuera de línea que crea un modelo aproximado de cada sistema de búsqueda especializado en un tópico. En su componente en línea, *Q-Pilot* rutea dinámicamente las consultas de los usuarios a los sistemas de búsquedas especializados.

- [Sei83] SEIDMAN, S. B. Network structure and minimum degree. *Social Networks* 5, 3 (1983), 269–287.

Se propone un enfoque para redes cohesivas que se basa en el mínimo grado y que produce una serie de sub-grafos con cohesión incremental. Además, este enfoque es útil tanto para la caracterización de las estructuras de red como para la comparación de las mismas.

- [Sha06] SHARAN, A. *Exploiting semantic locality to improve peer-to-peer search mechanisms*. PhD thesis, Rochester Institute of Technology, 2006.

En esta Tesis se propone un nuevo mecanismo para identificar contenido en sistemas P2P. Este sistema se basa en la localidad de los datos para rutear

las consultas de manera eficiente. Para evaluar la eficiencia de este modelo se implementó *LocalChord* que es un re-modelado de *Chord* en sub-Chords.

- [SMK⁺01] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review* 31, 4 (2001), 149–160.

En este artículo se presenta *Chord*, un protocolo para solucionar el problema de localizar nodos que almacenan un determinado ítem en un red P2P. Se basa en un método de mapeo entre claves y nodos. Este sistema funciona correctamente aún en redes en constante cambio, es decir con muchas llegadas y partidas de nodos. Se presentan resultados que han demostrado la escalabilidad de *Chord*.

- [SMwW⁺03] SUEL, T., MATHUR, C., WEN WU, J., ZHANG, J., DELIS, A., KHARRAZI, M., LONG, X., AND SHANMUGASUNDARAM, K. Odissea: A peer-to-peer architecture for scalable web search and information retrieval. In *International Workshop on the Web and Databases* (2003), pp. 67–72.

En este artículo se describe un prototipo de un sistema de búsqueda P2P para grandes colecciones de documentos. Este sistema es llamado *ODISSEA* y provee un gran índice global distribuido y un servicio de ejecución de consultas que puede utilizarse para localizar contenido dentro o fuera de la red P2P. Se presenta una descripción del sistema propuesto y una discusión respecto a cuestiones que se deberán afrontar al momento de su implementación.

- [SMZ03] SRIPANIDKULCHAI, K., MAGGS, B., AND ZHANG, H. Efficient content location using interest-based locality in peer-to-peer systems. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies* (2003), vol. 3, IEEE, pp. 2166–2176.

En este artículo se discute respecto a cómo mejorar el algoritmo de inundación o *flooding* utilizado por la aplicación *Gnutella* sin perder su simplicidad. Se propone una solución en la que los pares se organizan entre sí mismos en una estructura de acuerdo a su contenido. De esta manera, las búsquedas se realizan en esta estructura utilizando un algoritmo de inundación. Para concluir, se presentan y discuten los resultados experimentales.

- [SP04] SCHMIDT, C., AND PARASHAR, M. A peer-to-peer approach to web service discovery. *World Wide Web* 7, 2 (2004), 211–229.

En este artículo se presenta un sistema de indexado P2P. Este sistema brinda facilidades para almacenar gran cantidad de información en forma descentralizada y posee capacidad para realizar búsquedas en tiempo real. Este sistema soporta consultas complejas, tales como aquellas que poseen pocas palabras claves que son de gran utilidad al momento de hallar una respuesta. Los autores presentan el diseño del sistema y resultados de evaluaciones experimentales.

- [SR09] SIMARI, G. R., AND RAHWAN, I., Eds. *Argumentation in Artificial Intelligence*. Springer, 2009.

En este libro se presentan las tendencias, desarrollo y aplicaciones relacionadas a la Inteligencia Artificial (IA). En primer lugar se sitúan las contribuciones en un contexto histórico y luego se discute un gran número de temas que han emergido resultado de la aplicación de la argumentación en distintas áreas.

- [SSDN02] SCHLOSSER, M., SINTEK, M., DECKER, S., AND NEJDL, W. A scalable and ontology-based p2p infrastructure for semantic web services. In *Peer-to-Peer Computing, 2002.(P2P 2002). Proceedings. Second International Conference on* (2002), IEEE, pp. 104–111.

Los autores proponen sobrellevar los problemas de escalabilidad y congestión en redes P2P mediante una topología que permita el *broadcast* y la búsqueda eficiente. Además, presentan un algoritmo para la construcción de esta topología que no requiere de ningún servidor central ni de pares especiales o *super-peers*. Por último, los autores muestran cómo este enfoque puede ser mejorado mediante ontologías para determinar la organización de los pares en la topología.

- [SZTS07] SHOKOUHI, M., ZOBEL, J., TAHAGHOGHI, S., AND SCHOLER, F. Using query logs to establish vocabularies in distributed information retrieval. *Information Processing and Management* 43, 1 (2007), 169–180.

En este artículo se introducen dos aplicaciones de los *logs de consultas* en el contexto de recuperación de información en entornos distribuidos. En primer lugar se utilizan los términos de los logs para guiar el muestreo de colecciones distribuidas no cooperativas. En segundo lugar, se propone y evalúa una es-

trategia de poda que utiliza la información de los logs para la eliminación de términos.

- [TCM⁺11] TANG, Y., CAI, K., MCBURNEY, P., SKLAR, E., AND PARSONS, S. Using argumentation to reason about trust and belief. *Journal of Logic and Computation* 22, 5 (2011), 979–1018.

En este artículo se introduce un sistema formal de argumentación que se puede usar para razonar usando información referida a confiabilidad. El sistema se describe como un conjunto de grafos que hacen posible combinar este enfoque con enfoques convencionales para la representación de la confiabilidad. Se presentan algunas de las propiedades de estos grafos, las formas en las que se puede computar la confiabilidad y creencias y se ilustran todos estos conceptos mediante un ejemplo.

- [TGK⁺08] TONI, F., GRAMMATIKOU, M., KAFETZOGLOU, S., LYMBEROPOULOS, L., PAPAVALASSILEIOU, S., GAERTNER, D., MORGE, M., BROMURI, S., MCGINNIS, J., STATHIS, K., CURCIN, V., GHANEM, M., AND GUO, L. The argugrid platform: An overview. In *Grid Economics and Business Models*, J. Altmann, D. Neumann, and T. Fahringer, Eds., vol. 5206 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 217–225.

- [THI⁺10] TIRADO, J. M., HIGUERO, D., ISAILA, F., CARRETERO, J., AND IAMNITCHI, A. Affinity P2P: A self-organizing content-based locality-aware collaborative peer-to-peer network. *Computer Networks* 54, 12 (2010), 2056–2070.

En este artículo se presenta el sistema *Affinity P2P* (AP2P). Es un sistema P2P basado en clusters y en auto-organización. Además, se presenta una métrica denominada *affinity-based metric* para medir la distancia entre clústers y un algoritmo de búsqueda para hallar respuestas dentro de este sistema. Se presentan resultados de simulaciones y una extensiva evaluación de los mismos.

- [THT12] TIGELAAR, A. S., HIEMSTRA, D., AND TRIESCHNIGG, D. Peer-to-peer information retrieval: An overview. *ACM Trans. Inf. Syst.* 30, 2 (May 2012), 9:1–9:34.

En este artículo se brinda una revisión de los principales desafíos que se presentan a la hora de recuperar información en redes P2P. El objetivo de los autores es inspirar futuras investigaciones para sobrellevar estos desafíos.

- [Ton08] TONI, F. *Assumption-Based Argumentation for Selection and Composition of Services*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 231–247.

En este artículo se presenta un enfoque basado en la argumentación para diseñar e implementar agentes que soporten selección y composición de servicios en entornos distribuidos. Los agentes se encuentran equipados con creencias, metas y decisiones alternativas. En este enfoque se resuelven los conflictos y preferencias usando el framework argumentativo ABA.

- [Ton12] TONI, F. Reasoning on the web with assumption-based argumentation. In *Reasoning Web* (2012), Springer, pp. 370–386.

En este artículo se presenta un tutorial respecto a la argumentación en el área de la inteligencia artificial. Se enfoca en la argumentación abstracta y en la argumentación basada en presunciones. Se discuten posibles usos en contextos web, particularmente en la web semántica y en redes sociales.

- [Ton13] TONI, F. A generalised framework for dispute derivations in assumption-based argumentation. *Artificial Intelligence* 195 (2013), 1–43.

- [Ton14] TONI, F. A tutorial on assumption-based argumentation. *Argument and Computation* 5 (2014), 89–117.

En este artículo su autora presenta una descripción del framework ABA junto con sus principales definiciones. Además, presenta un gran número de ejemplos respecto al comportamiento del framework bajo distintas situaciones y para concluir presenta una sección de preguntas frecuentes con respuestas muy claras y explicativas.

- [TR03] TSOUMAKOS, D., AND ROUSSOPOULOS, N. Adaptive probabilistic search for peer-to-peer networks. In *Peer-to-Peer Computing, 2003.(P2P 2003). Proceedings. Third International Conference on* (2003), IEEE, pp. 102–109.

En este artículo se presenta un algoritmo adaptativo y de fácil desarrollo para ejecutar búsquedas en redes P2P no estructuradas. Este método se llama

Adaptive Probabilistic Search method (APS) y utiliza experiencia pasada para rutear futuras consultas. Se presentan distintas simulaciones y los resultados obtenidos mediante las mismas.

- [TSW04] TEMPICH, C., STAAB, S., AND WRANIK, A. Remindin': Semantic query routing in peer-to-peer networks based on social metaphors. In *Proceedings of the 13th International Conference on World Wide Web* (New York, NY, USA, 2004), WWW '04, ACM, pp. 640–649.

En este artículo se propone un método llamado *REMINDIN'* que utiliza metáforas sociales para resolver el problema de la búsqueda eficiente en sistemas P2P. Se presentan y analizan resultados de simulaciones de este algoritmo y de algunas variantes.

- [twi17] Twister. <http://twister.net.co/>. Recuperado en Junio de 2017, 2017.
- Twister* es una plataforma P2P para *microblogging* creada a partir del código libre de las aplicaciones *Bitcoin* y *BitTorrent*.

- [TXM03] TANG, C., XU, Z., AND MAHALINGAM, M. psearch: Information retrieval in structured overlays. *SIGCOMM Comput. Commun. Rev.* 33, 1 (January 2003), 89–94.

En este artículo se presenta *pSearch*, un sistema de recuperación de información en redes P2P que mejora los problemas de escalabilidad presentes en las búsquedas centralizadas o la ineficiencia presentada por los algoritmos estilo *Flooding*. En este sistema los documentos son organizados de acuerdo a su representación vectorial, de este modo el espacio de búsqueda para una dada consulta es organizado en torno a documentos relacionados.

- [UAO13] UEDA, K., AKASE, J.-I., AND OKUBO, T. Analysis of peer cluster layers selection criteria for p2p contents distribution systems. In *APNOMS* (2013), pp. 1–6.

En este artículo se presenta una extensión de un estudio previo en el que el autor proponía una arquitectura para construir clusters en múltiples capas para mejorar el rendimiento de las búsquedas en sistemas P2P. En este nuevo artículo se incluye un estudio respecto a cómo manejar contenidos populares para la formación de los clusters.

- [UO12] UEDA, K., AND OKUBO, T. Peer-to-peer contents distribution system using multiple peer clusters. In *2012 14th Asia-Pacific Network Operations and Management Symposium (APNOMS)* (2012).

Los autores presentan una arquitectura para construir clusters en múltiples capas para mejorar el rendimiento de las búsquedas en sistemas P2P. La idea es minimizar la distancia física entre clusters formados en forma lógica, de este modo se evita tener un gran número de nodos intermediarios para el re-envío de los mensajes.

- [VH08] VARGA, A., AND HORNIG, R. An overview of the omnet++ simulation environment. In *1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops* (Brussels, Belgium, 2008), Simutools '08, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 60:1–60:10.

Este artículo presenta el framework *OMneT++* y trabajos sobre el mismo para su actualización. Este simulador se encuentra disponible desde 1997. Se creó para simular redes de comunicación, multiprocesos y otros tipos de sistemas distribuidos. En lugar de crear un simulador específico para cada área, *OMneT++* fue diseñado para ser lo mas general posible. Este simulador ha sido utilizado en un gran número de dominios, entre ellos simulaciones de redes P2P.

- [VKMvO04] VOULGARIS, S., KERMARREC, A., MASSOULIÉ, L., AND VAN OTEEN, M. Exploiting semantic proximity in peer-to-peer content searching. In *Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 238–243.

En este artículo se ataca la problemática de la búsqueda de contenido en sistemas P2P modificando la topología lógica que es la que describe las relaciones entre los pares miembros del sistema. Se propone crear una red semántica donde aquellos nodos similares se encuentren próximos en distancia entre sí. Se muestran algunos enfoques para calcular la proximidad semántica, y se presenta una comparativa de los mismos.

- [WAM05] WU, L.-S., AKAVIPAT, R., AND MENCZER, F. 6s: Distributing crawling and searching across web peers. In *Web Technologies, Applications, and Services* (2005), pp. 159–164.
- Los autores presentan *6Search(6S)*, una aplicación P2P colaborativa que considera las limitaciones de escalabilidad que aparecen en los sistemas centralizados. Los pares o miembros de este sistema dependen de un algoritmo adaptativo local que permite cambiar dinámicamente la topología para enviar las consultas hacia los pares apropiados. Se realizaron distintas simulaciones y los resultados producto de las mismas fueron comparados con sistemas de búsqueda centralizados como Google.
- [WBS08] WALTER, F. E., BATTISTON, S., AND SCHWEITZER, F. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems* 16, 1 (2008), 57–74.
- En este artículo los autores presentan un modelo basado en confianza para sistemas de recomendación en redes sociales. La idea de este modelo es que los agentes usen sus redes sociales para recopilar información y crear relaciones de confianza. Se presenta un estudio respecto a cómo se modifican estas relaciones a medida que un agente se vuelve mas o menos confiable. Además, se presenta un análisis de cómo afecta la densidad de la red y la heterogeneidad de sus agentes.
- [WS98] WATTS, D. J., AND STROGATZ, S. H. Collective dynamics of ‘small-world’ networks. *Nature* 393, 6684 (1998), 440–442.
- En este artículo se exploran distintos modelos topológicos de redes. En particular se presenta y define el concepto de *topología de mundo pequeño*. Se demuestra que hay numerosas redes de la vida real que presentan este tipo de topología. Por ejemplo, redes neuronales, redes de interacción entre proteínas, red de colaboración entre actores y películas, etc.
- [WW13] WU, K., AND WU, C. State-based search strategy in unstructured p2p. *Future Generation Computer Systems* 29, 1 (2013), 381–386.
- En este artículo se presenta un algoritmo de búsqueda en redes P2P llamado *State-Based Search* (SBS). Este algoritmo utiliza información respecto al estado

de los nodos para tomar sus decisiones. Para corroborar la correctitud de este algoritmo se presentan resultados de simulaciones en entornos dinámicos. Los resultados que se presentan demuestran que este algoritmo logra reducir el tiempo de respuesta y lograr un mejor balance de carga.

- [YDRC06] YANG, Y., DUNLAP, R., REXROAD, M., AND COOPER, B. F. Performance of full text search in structured and unstructured peer-to-peer systems. In *In IEEE INFOCOM* (2006), IEEE Press.

Los autores presentan una comparación cuantitativa respecto a búsquedas de texto y palabras claves en sistemas P2P estructurados y no estructurados. En particular examinan tres técnicas populares en la literatura y sus optimizaciones. La primera de estas técnicas combina el uso de tablas hash distribuidas y *Bloom-filters*; la segunda utiliza el concepto de *super-par* y la última utiliza un algoritmo estilo *random walk* en redes no estructuradas. Presentan simulaciones realistas y una evaluación de los resultados obtenidos de las mismas.

- [YGM02] YANG, B., AND GARCIA-MOLINA, H. Improving search in peer-to-peer networks. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on* (2002), IEEE, pp. 5–14.

En este artículo se presentan tres técnicas para efectuar búsquedas en sistemas P2P. Se presenta el diseño de estas técnicas junto con sus evaluaciones. Las técnicas que se proponen son sencillas, es por ello que su integración a sistemas ya existentes no debería implicar un gran desafío.

- [YGS15] YU, Y.-T., GERLA, M., AND SANADIDI, M. Scalable vanet content routing using hierarchical bloom filters. *Wireless Communications and Mobile Computing* 15, 6 (2015), 1001–1014.

En este artículo se discute un método de ruteo orientado al contenido que permite almacenar, compartir y buscar la totalidad de los datos dentro de las *vehicular ad-hoc networks* (VANET). Los autores introducen un esquema pro-activo para descubrir contenido llamado *Hierarchical Bloom-Filter Routing* (HBFR) que permite abordar distintas cuestiones de las VANET tales como la movilidad y la gran cantidad de datos.

- [YL13] YU, W., AND LIN, X. Irwr: incremental random walk with restart. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* (2013), ACM, pp. 1017–1020.
- Este artículo presenta un enfoque para computar la métrica *Random Walk with Restart* (RWR) en grafos dinámicos. Este tipo de grafos son grafos de tamaño considerable los cuales se actualizan con pequeñas modificaciones. Los autores proponen un algoritmo rápido y exacto para calcular el RWR en grafos con actualizaciones de arcos.
- [YXC⁺15] YANG, Z., XING, Y., CHEN, C., XUE, J., AND DAI, Y. Understanding the performance of offline download in real p2p networks. *Peer-to-Peer Networking and Applications* 8, 6 (2015), 992–1007.
- En este artículo se presenta un enfoque para mejorar el rendimiento de las descargas sin conexión (offline) en sistemas P2P. Mediante la realización de modelos y su verificación experimental bajo circunstancias realistas pudieron confirmar la viabilidad y efectividad del sistema propuesto para la descarga de archivos sin conexión para distintos sistemas P2P..
- [YZ04] YAN, F., AND ZHAN, S. A peer-to-peer approach with semantic locality to service discovery. In *Grid and Cooperative Computing - GCC 2004*, H. Jin, Y. Pan, N. Xiao, and J. Sun, Eds., vol. 3251 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 831–834.
- [ZHF10] ZHU, Y., HU, R., AND FEI, L. A low latency resource location algorithm for unstructured p2p networks. In *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on* (2010), IEEE, pp. 1–4.
- En este artículo se presenta un enfoque para construir un servicio de localización de recursos de baja latencia en redes P2P no estructuradas. Este enfoque propone seleccionar los mejores caminos para enviar las consultas dependiendo de las características de los nodos. Se presentan simulaciones y evaluaciones de los resultados que demuestran que este método se adapta a cambios en la topología y en el estado de las conexiones..

- [ZHS⁺04] ZHAO, B. Y., HUANG, L., STRIBLING, J., RHEA, S. C., JOSEPH, A. D., AND KUBIATOWICZ, J. D. Tapestry: A resilient global-scale overlay for service deployment. *Selected Areas in Communications, IEEE Journal on* 22, 1 (2004), 41–53.

Los autores presentan *Tapestry*, una infraestructura P2P para el ruteo eficiente y escalable de consultas. Este método soporta la API de localización descentralizada de objetos y ruteo (DOLR). En este artículo se presenta la arquitectura del sistema propuesto junto con sus algoritmos e implementaciones. Se presentan simulaciones realizadas con la herramienta *PlanetLab* y la evaluación de los datos obtenidos mediante las mismas.

- [ZI16] ZUO, X., AND IAMNITCHI, A. A survey of socially aware peer-to-peer systems. *ACM Computing Surveys (CSUR)* 49, 1 (2016), 9.

En este artículo se presenta un trabajo de revisión de literatura respecto a la información social que ha sido utilizada para el diseño de sistemas P2P. Además, se provee una clasificación de esta información y se discuten sus usos y desafíos en este tipo de sistemas.

- [ZLF⁺05] ZHUGE, H., LIU, J., FENG, L., SUN, X., AND HE, C. Query routing in a peer-to-peer semantic link network. *Computational Intelligence* 21, 2 (2005), 197–216.

En este artículo se presenta un enfoque que permite descubrir enlaces en redes P2P. Es una herramienta que permite crear y mantener enlaces semánticos que utiliza una medida de similitud para el ruteo eficiente de consultas. En el enfoque que se propone se pueden distinguir tres aspectos importantes. Primero, el uso de enlaces semánticos para enriquecer las relaciones entre pares. Segundo, considera no sólo los nodos sino que una estructura XML para medir la similitud entre esquemas, lo que permite rutear acertadamente las consultas hacia pares potencialmente útiles. En tercer y último lugar este enfoque permite trabajar con datos heterogéneos y posiblemente inconsistentes.

- [ZWH03] ZHU, Y., WANG, H., AND HU, Y. Integrating semantics-based access mechanisms with p2p file systems. In *Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on* (Sept 2003), pp. 118–125.

En este artículo se presentan las bases de una arquitectura para sistemas de archivos P2P que soporta acceso basado en semántica. Su contribución principal es el uso de la semántica de los datos para indexar y ubicar información. El enfoque utiliza tablas hash distribuidas donde los índices de archivos con semántica similar se encuentran agrupados entre los mismos pares. De este modo, una consulta puede encontrar una respuesta consultando sólo a un pequeño número de pares. Este enfoque sólo agrega información respecto a índices a cada nodo, lo que no provoca sobrecarga alguna a nivel de almacenamiento.

- [ZYKG04] ZEINALIPOUR-YAZTI, D., KALOGERAKI, V., AND GUNOPULOS, D. Information retrieval techniques for peer-to-peer networks. *Computing in Science Engineering* 6, 4 (2004), 20–26.

Los autores presentan un resumen del estado del arte respecto a técnicas para la recuperación de información en redes P2P, incluyendo una evaluación experimental realista y una comparativa de las técnicas descriptas.